On the Total Power Capacity of Regular-LDPC Codes with Iterative Message-Passing Decoders

Karthik Ganesan†, Pulkit Grover‡, Jan Rabaey§, and Andrea Goldsmith†

***Abstract*—Motivated by recently derived fundamental limits on total (transmit + decoding) power for coded communication with VLSI decoders, this paper investigates the scaling behavior of the minimum total power needed to communicate over AWGN channels as the target bit-error-probability tends to zero. We focus on regular-LDPC codes and iterative message-passing decoders. We analyze scaling behavior under two VLSI complexity models of decoding. One model abstracts power consumed in processing elements ("node model"), and another abstracts power consumed in wires which connect the processing elements ("wire model"). We prove that a coding strategy using regular-LDPC codes with Gallager-B decoding achieves order-optimal scaling of total power under the node model. However, we also prove that regular-LDPC codes and iterative message-passing decoders cannot meet existing fundamental limits on total power under the wire model. Further, if the transmit energy-per-bit is bounded, total power grows at a rate that is *worse* than uncoded transmission. Complementing our theoretical results, we develop detailed physical models of decoding implementations using post-layout circuit simulations. Our theoretical and numerical results show that approaching fundamental limits on total power requires increasing the complexity of both the code design and the corresponding decoding algorithm as communication distance is increased or error-probability is lowered.**

***Index Terms*—Low-density parity-check (LDPC) codes; Iterative message-passing decoding; Total power channel capacity; Energy-efficient communication; System-level power consumption; Circuit power consumption; VLSI complexity theory.**

I. Introduction

Intuitively, the concept of Shannon capacity captures how much information can be communicated across a channel under specified resource constraints. While the problem of approaching Shannon capacity under solely *transmit power* constraints is well understood, modern communication often takes place at transmitter-receiver distances that are very short (e.g., on-chip communication [3], short distance wired communication [4], and extremely-high-frequency short-range wireless communication [5]). Empirically, it has been observed that at such short distances, the power required for processing a signal at the transmitter/receiver circuitry can dominate the power required for transmission, sometimes by orders of magnitude [4], [6], [7]. For instance, the power consumed in the decoding circuitry of multi-gigabit-per-second communication systems can be hundreds of milliwatts or more (e.g., [4], [8]), while the transmit power required is only tens of milliwatts [7]. Thus, transmit power constraints do not abstract the relevant power consumed in many modern systems.

Shannon capacity, complemented by modern coding-theoretic constructions [9], has provided a framework that is provably good for minimizing transmit power (e.g., in power-constrained AWGN channels). In this work, we focus on a capacity question that is motivated by *total power*: at what maximum rate can one communicate across a channel for a given total power, and a specified error-probability? Alternatively, given a target communication rate and error-probability, what is the minimum required total power? The first simplifying perspective to this problem was adopted in [10], [11], where all of the processing power components at the transmitter and the receiver were lumped together. However, processing power is influenced heavily by the specific modulation choice, coding strategy, equalization strategy, etc. [4], [6]. Even for a fixed communication strategy, processing power depends strongly on the implementation technology (e.g., 45 nm CMOS) and the choice of circuit architecture.

Using theoretical models of VLSI implementations [12], recent literature has explored fundamental scaling limits [6], [13], [14], [15] on the transmit + decoding power consumed by error-correcting codes. These works abstract energy consumed in processing nodes [6] and wires [14], [13], [15] in the VLSI decoders, and show that there is a fundamental tradeoff between transmit and decoding power.

In this work, we examine the achievability side of the question (see Fig. 1): what is the total power that known code families and decoding algorithms can achieve? To address this question, we first provide asymptotic bounds (Sections IV-V) on required decoding power. To do so, we restrict our analysis to binary regular-LDPC codes and iterative message-passing decoding algorithms. Our code-family choice is motivated by both the order-optimality of regular-LDPC codes in some theoretical models of circuit power [6], and their practical utility in both short [16] and long [17] distance settings. Recent work of Blake and Kschischang [18] also studied the energy complexity of LDPC decoding circuits, and an important connection to this paper is highlighted in Section VII.

Within these restrictions we provide the following insights:

1) Wiring power, which explicitly brings out physical constraints in a digital system [19], costs more in the order sense than the power consumed in processing nodes. Thus, the commonly used metric for decoding complexity — number of operations — underestimates circuit energy costs.
2) Shannon capacity is the maximal rate one can communicate at with arbitrary reliability while the transmit power

† Electrical Engineering, Stanford University. ‡ Electrical and Computer Engineering, Carnegie Mellon University. § Electrical Engineering and Computer Science, University of California at Berkeley. (Email correspondence should be addressed to karthik3@stanford.edu.)

Early results related to this paper were presented at the 2012 Allerton Conference [1] and IEEE Globecom 2012 [2].

is held *fixed*. However, when total power minimization is the goal, keeping transmit power fixed while bit-error probability approaches zero can lead to highly suboptimal decoding power. For instance, we prove that (Theorems 3, 4, 5) at sufficiently low bit-error probability, it is more total power efficient to use *uncoded transmission* than regular-LDPC codes with iterative message-passing decoding, if using fixed transmit power. However, if transmit power is allowed to diverge to infinity, we show that regular-LDPC codes can outperform uncoded transmission in this total power sense.

3) We prove (Corollary 2) that a strategy using regular-LDPC codes and the Gallager-B decoding achieves order-optimal scaling of total power when processing power is dominated by nodes as opposed to wires (see Section IV-C).
4) However, we also prove a lower bound (Theorem 3) that holds for all regular-LDPC codes with iterative message-passing decoders for the case where processing power is dominated by wires, and we show that a large gap exists between this lower bound and existing fundamental limits (see Section V-C).

To obtain insights on how an engineer might choose a power-efficient code for a given system, we then develop empirical models of decoding power consumption of 1-bit and 2-bit message-passing algorithms for regular-LDPC codes (Section VI-C). These models are constructed using post-layout circuit simulations of power consumption for check-node and variable-node sub-circuits, and generalizing the remaining components of power to structurally similar codes.

Shannon-theoretic analysis yields transmit-power-centric results, which are plotted as "waterfall" curves (with corresponding "error-floors") demonstrating how close the code performs to the Shannon limit. There, the channel path-loss can usually be ignored because it is merely a scaling factor for the term to be optimized (namely the transmit power), thereby not affecting the optimizing code. Since we are interested in *total* power, the path-loss impacts the code choice. For simplicity of understanding, path-loss is translated into a more relatable metric — communication distance — using a simple model for path-loss. The resulting question is illustrated in Fig. 1(b): At a given data-rate, what code and corresponding decoding algorithm minimize the transmit + decoding power for a given transmit distance and bit-error probability?

In Section VI-C, we present optimization results for this question in a 60 GHz communication setting using our models. This particular setting is chosen not just because of the short distance, but also because the results highlight another conceptual point we stress in this paper:

5) Approaching total power capacity requires an increase in the complexity of both the code design and the corresponding decoding algorithm as communication distance is increased, or bit-error probability is lowered.

The results presented in this paper have some limitations. First, we only consider a limited set of coding strategies, and while the results and models presented here extend easily to irregular LDPC constructions, they are not necessarily applicable to all decoders. Second, modern transceivers [20] contain many other processing power sinks, including analog-to-digital converters (ADCs), digital-to-analog converters (DACs), power amplifiers, modulation, and equalizers, and the power requirements of each of these components can vary[1] based on the coding strategy. While recent works have started to address fundamental limits [21] and modeling [22] of power consumption of system blocks from a mixed-signal circuit design perspective, tradeoffs with *code choice* of these components remain relatively unexplored. Hence, while analyzing decoding power is a start, other system-level tradeoffs should be addressed in future work. It is also of great interest to understand tradeoffs at a network level (see [6]), where multiple transmitting-receiving pairs are communicating in a shared wireless medium. In such situations, one cannot simply increase transmit power to reduce decoding power: the resulting interference to other users needs to be accounted for as well.

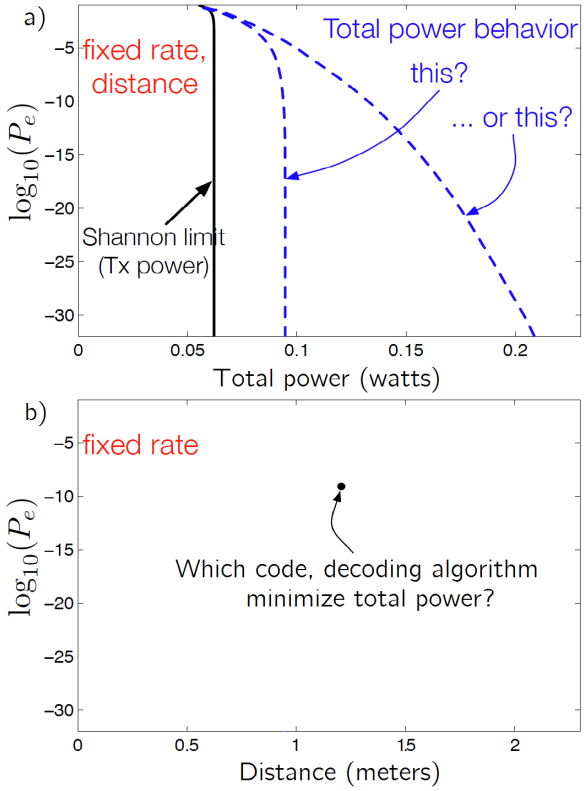


Fig. 1: a). The question explored in Sections II-V: How fast does total power diverge to ∞ as bit-error probability $P_e \to 0$ for regular-LDPC codes and iterative message-passing decoding algorithms? b). The question explored in Section VI-C: what is the most power-efficient pairing of a code and decoding algorithm for a given distance and bit-error probability?

The remainder of the paper is organized as follows. Section II states the assumptions and notation used in the paper. Sections II-C to II-G introduce theoretical models of VLSI circuits and decoding energy. Preliminary results are stated in Section III, which are used to analyze decoding energy in

[1]For example, the *resolution* of ADCs used at the receiver may vary with the code choice by virtue of the fact that changing the *rate* of the code may require a change in signaling constellation (when channel bandwidth and data-rate are fixed).

Sections IV and V, in the context of the question illustrated in Fig. 1a) (obtaining the scaling behavior). Section VI discusses circuit-simulation-based numerical models of decoding power, in the context of the question illustrated in Fig. 1b). Section VII concludes the paper.

II. System and VLSI models for asymptotic analysis

Throughout this paper, we rely on Bachmann-Landau notation [23] (i.e. "big-O" notation). We first state a preliminary definition that is needed in order to state a precise definition of the big-O notation that we use in this paper.

Definition 1. $\mathcal{X} \subseteq \mathbb{R}$ is a right-sided set if $\forall x \in \mathcal{X}$, $\exists y \in \mathcal{X}$ such that $y > x$.

Some examples of right-sided sets include $\mathbb{R}$, $\mathbb{N}$, and intervals of the form $[a, \infty)$, where a is a constant. We now state the Bachmann-Landau notation for non-negative real-valued functions defined on right-sided sets[2].

Definition 2. Let $f : \mathcal{X} \to \mathbb{R}^{\geq 0}$ and $g : \mathcal{X} \to \mathbb{R}^{\geq 0}$ be two non-negative real-valued functions, both defined on a right-sided set $\mathcal{X}$. We state

1) $f(x) = \mathcal{O}(g(x))$ if $\exists x_1 \in \mathcal{X}$ and $c_1 > 0$ s.t. $f(x) \leq c_1 g(x)$, $\forall x \geq x_1$.
2) $f(x) = \Omega(g(x))$ if $\exists x_2 \in \mathcal{X}$ and $c_2 > 0$ s.t. $f(x) \geq c_2 g(x)$, $\forall x \geq x_2$.
3) $f(x) = \Theta(g(x))$ if $\exists x_3 \in \mathcal{X}$ and $c_4 \geq c_3 > 0$ s.t. $c_3 g(x) \leq f(x) \leq c_4 g(x)$, $\forall x \geq x_3$.

We will also need a Bachmann-Landau notation for *two* variable functions [24, Section 3.5]:

Definition 3. Let $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{\geq 0}$ and $v : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{\geq 0}$ be two non-negative real-valued functions, both defined on the Cartesian product of two right-sided sets $\mathcal{X}$ and $\mathcal{Y}$. We state

1) $u(x, y) = \mathcal{O}(v(x, y))$ if $\exists M \in \mathbb{R}$ and $c_1 > 0$ s.t. $u(x, y) \leq c_1 v(x, y)$, $\forall x, y \geq M$.
2) $u(x, y) = \Omega(v(x, y))$ if $\exists M \in \mathbb{R}$ and $c_2 > 0$ s.t. $u(x, y) \geq c_2 v(x, y)$, $\forall x, y \geq M$.
3) $u(x, y) = \Theta(v(x, y))$ if $\exists M \in \mathbb{R}$ and $c_4 \geq c_3 > 0$ s.t. $c_3 v(x, y) \leq u(x, y) \leq c_4 v(x, y)$, $\forall x, y \geq M$.

We will often apply Definitions 2 and 3 in the limit as bit-error probability $P_e \to 0$, where the definitions can be interpreted as applied to a function with an argument $\frac{1}{P_e}$ as it diverges to ∞. All logarithm functions $\log(\cdot)$ are natural logarithms unless otherwise stated.

A. Communication channel model

We assume the communication between transmitter and receiver takes place over an AWGN channel with fixed attenuation. The transmission strategy uses BPSK modulation, and a (d_v, d_c)-regular binary LDPC code of design rate $R = 1 - \frac{d_v}{d_c}$ [25] (which is assumed to equal the code rate). The blocklength of the code is denoted by n, and the number of source bits is denoted by $k = nR$. The decoder performs a hard-decision on the observed channel outputs before starting the decoding process, thereby first recovering noisy codeword bits transmitted through a Binary Symmetric Channel (BSC) of flip probability $p_0 = \mathbb{Q}\left(\sqrt{2\frac{E_s}{N_0}}\right)$. Here, E_s is the input energy per channel symbol and $\frac{N_0}{2}$ is the noise power. $\mathbb{Q}(\cdot)$ is the tail probability of the standard normal distribution, $\mathbb{Q}(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-u^2}{2}}\, du$. The transmit power P_T is assumed to be proportional to $\frac{E_s}{N_0}$, modeling fixed distance and constant attenuation wireless communication. Explicitly we assume $\frac{E_s}{N_0} = \eta P_T$ for some constant $\eta > 0$. Using known bounds on the $\mathbb{Q}$-function [26], $\frac{e^{-x^2/2}}{\sqrt{2\pi}(x+1/x)} \leq \mathbb{Q}(x) \leq \frac{e^{-x^2/2}}{\sqrt{2\pi}x}$:

$$\frac{e^{-\eta P_T}}{\sqrt{4\pi\eta P_T} + \sqrt{\frac{\pi}{\eta P_T}}} \leq p_0 \leq \frac{e^{-\eta P_T}}{\sqrt{4\pi\eta P_T}}. \tag{1}$$

The focus of this paper is on the analysis of the "total" power required to communicate on the above channel, as the target average bit-error probability $P_e \to 0$. Our simplified notion of total power is defined below.

Definition 4. The total power, P_{total}, consumed in communication across the channel described in II-A is defined as

$$P_{\text{total}} = P_T + P_{\text{Dec}}, \tag{2}$$

where P_T and P_{Dec} are the power spent in transmission and decoding, respectively.

The channel model helps analyze the transmit power component in (2), but a model for decoding power is also needed. In the next section, we provide models and assumptions for decoding algorithms and implementations that are used in the paper. We allow P_T and P_{Dec} to be chosen depending on P_e, η, and the coding strategy. Throughout the paper, the minimum total power for a strategy is denoted by $P_{\text{total,min}}$ and the optimizing transmit power by P_T^*.

B. Decoding algorithm assumptions

The general theoretical results of this paper (Lemma 2, Theorem 3) hold for any iterative message-passing decoding algorithm (and *any* number of decoding iterations) that satisfies "symmetry conditions" in [25, Def. 1] (which allow us to assume that all-zero codeword is transmitted). Thus, each node only operates on the messages it receives at its inputs. We note that the sum-product algorithm [27], the min-sum algorithm [28], Gallager's algorithms [29], and most other message-passing decoders satisfy these assumptions. For the constructive results of this paper (Corollary 1, Corollary 2, Theorem 4, Theorem 5) we focus on the two decoding algorithms originally proposed in Gallager's thesis [29], that are now called "Gallager-A" and "Gallager-B" [25]. For these results, we will use density-evolution analysis [9] to analyze the performance[3], for which we define the term "independent

[2]Bounded intervals that are open on the right such as $(-1, 0)$ or $[0, 5)$ are also right-sided sets. Definition 2 can still be applied to functions restricted to such sets, but we will not consider such functions in this paper.

[3]In practice, decoding is often run for a larger number of iterations because at large blocklengths, bit-error probability may still decay as the number of iterations increase. In that case, density-evolution does not yield the correct bit-error probability, as it will vary based on the code construction [30].

iterations" as follows:

Definition 5. An *independent decoding iteration* is a decoding iteration in which messages received at a single variable or a check node are mutually independent.

We will denote number of independent iterations[4] that an algorithm runs as N_{iter}. This quantity is constrained by the *girth* [25] of the code, defined as the length of the shortest cycle in the Tanner graph of the code [31] as follows: for a code with girth g, the maximum value of N_{iter} is $\lfloor \frac{g-2}{4} \rfloor$.

C. VLSI model of decoding implementation

Theoretical models for analyzing area and energy costs of VLSI circuits were introduced several decades ago in computer science. These include frameworks such as the Thompson [12] and Brent-Kung [32] models for circuit area and energy complexity (called the "VLSI models"), and Rent's rule [33], [34]. Our model for the LDPC decoder implementation in this paper is an adaptation of Thompson's model [12], and it entails the following assumptions:

1) The VLSI circuit includes processing nodes which perform computations and store data, and wires which connect them. The circuit is placed on a square grid of horizontal and vertical wiring tracks of finite width $\lambda > 0$, and contact squares of area λ^2 at the overlaps of perpendicular tracks.
2) Neighboring parallel tracks are spaced apart by width λ.
3) Wires carry information bi-directionally. Distinct wires can only cross orthogonally at the contact squares.
4) The layout is drawn in the plane. In other words, the model does not allow for more than two metal layers for routing wires in the manner that modern IC manufacturing processes do (see Section II-G1).
5) The processing nodes in the circuit have finite memory and are situated at the contact squares of the grid. They connect to wires routed along the grid.
6) Since wires are routed only horizontally and vertically, any single contact has access to a maximum of 4 distinct wires. To accommodate higher-degree nodes, a processing element requiring x external connections (for $x > 4$) can occupy a square of side-length $x\lambda$ on the grid, with wires connecting to any side. No wires pass over the large square.

λ models the minimum feature-size which is often used to describe IC fabrication processes. We refer to this model as Implementation Model (λ). The decoder is assumed to be implemented in a "fully-parallel" manner [8], i.e. a processing node never acts as more than one vertex in the Tanner graph [31] of the code. Each variable-node and check-node of an LDPC code is therefore represented by a distinct processing node in the decoding circuit. As an example, Fig. 2(a) shows the Tanner graph for a $(7,4)$-Hamming code and Fig. 2(b) shows a fully-parallel layout of a decoder for the same code. In Sections II-E, II-F we will describe two models[5] of energy consumption for the VLSI decoder.

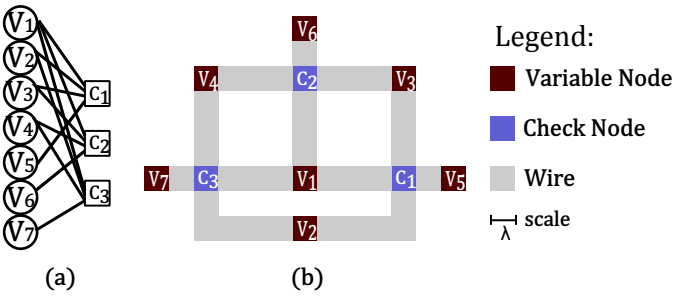


Fig. 2: The Tanner graph (a) of a $(7,4)$-Hamming code and a fully parallel decoder (b) drawn according to Implementation Model (λ). Each vertex in the Tanner graph corresponds to a processing node in the layout and each edge in the Tanner graph corresponds to a wire connecting distinct nodes.

D. Time required for processing

In order to translate the model of II-E to a power model, we need the time required for computation (the computation time is measured in seconds and is different from the number of algorithmic iterations). The computations are assumed to happen in clocked iterations, with each iteration consisting of two steps: passing of messages from variable to check nodes, and then from check to variable nodes. If the decoding algorithm requires the exchange of multi-bit messages, we assume the message bits can be passed using a single wire.

We denote the decoding throughput (number of source bits decoded per second) by R_{data}. Because a batch of k source bits are processed in parallel, the time available for processing is $T_{\text{proc}} = \frac{k}{R_{\text{data}}}$ seconds.

E. Processing node model of decoding power

Definition 6 (Node Model (ξ_{node}))**.** The energy consumed in each variable or check node during one decoding iteration is E_{node}. This constant can depend on λ, d_v and d_c. The total number of nodes at the decoder[6] is $n_{\text{nodes}} = n + (n-k) = 2n-k$. The total energy consumed in τ_{iter} decoding iterations is $E_{\text{nodes}} = E_{\text{node}} n_{\text{nodes}} \tau_{\text{iter}}$. The decoding power is $P_{\text{nodes}} = \frac{E_{\text{nodes}}}{T_{\text{proc}}} = \frac{E_{\text{node}}(2n-k)\tau_{\text{iter}}}{k} R_{\text{data}} = \xi_{\text{node}} \tau_{\text{iter}}$.

Here, $\xi_{\text{node}} = E_{\text{node}} \left(\frac{2}{R} - 1\right) R_{\text{data}} = \frac{E_{\text{node}}(d_v+d_c)}{(d_c-d_v)} R_{\text{data}}$. Note that τ_{iter} need not be tied to N_{iter}.

This model assumes that the entirety of the decoding energy is consumed in processing nodes, and wires require no energy. In essence, this model is simply counting the number of operations performed in the message-passing algorithm. The next energy model complements the node model by accounting for energy consumed in wiring.

F. Message-passing wire model of decoding power

Definition 7 (Wire Model (ξ_{wire}))**.** The decoding power is $P_{\text{wires}} = C_{\text{unit-area}} A_{\text{wires}} V_{supply}^2 f_{\text{clock}}$, where $C_{\text{unit-area}}$ is the capacitance per unit-area of a wire, V_{supply} is the supply

[4] In our constructive results, we constrain the decoder to only perform independent iterations. Thus, the number of independent iterations is the same as the number of iterations for those results, but it emphasizes on the requirement on the code to ensure that the girth is sufficiently large.

[5] Both models can also be used simultaneously. However, for simplicity, we present the results for the two models separately.

[6] In practice, many decoder implementations actually contain more than $n-k$ check nodes in order to break up small stopping-sets in the code. However, we do not consider such decoders in this paper.

voltage of the circuit, f_{clock} is the clock-frequency of the circuit, and A_{wires} is the total area occupied by the wires in the circuit. The parameters $C_{\text{unit-area}}$ and V_{supply} are technology choices that may depend on λ, d_v, and d_c. The parameter f_{clock} also may depend on λ, d_v, d_c, R_{data}, and the decoding algorithm. For simplicity, we write $\xi_{\text{wire}} = C_{\text{unit-area}} V_{supply}^2 f_{\text{clock}}$ and $P_{\text{wires}} = \xi_{\text{wire}} A_{\text{wires}}$.

Wires in a circuit consume power whenever they are "switched," i.e., when the message along the wire changes its value[7]. The probability of wire switching in a message-passing decoder depends on the statistics of the number of errors in the received word. These statistics depend on the flip-probability of the channel, which is controlled by the transmit power. Further, as decoding proceeds the messages also tend to stabilize, reducing switching and hence the power consumed in the wires. Activity-factor [19] could therefore be introduced in the Wire Model via a multiplicative factor between 0 and 1 that depends on P_T, η, and the decoding algorithm, but modeling it accurately would require a *very* careful analysis.

G. On modern VLSI technologies and architectures

The VLSI model of Section II-C and the assumptions made about the decoding architecture may seem pessimistic compared to the current state-of-the-art. However, in this section we justify our choices by explaining how many of the architecture and technology optimizations that are helpful in current practice have no impact on the conclusions derived by our theoretical analysis.

1) Multiple routing layers: Modern VLSI technologies allow for upwards of 10 metal layers for routing wires [35]. While this helps *significantly* in reducing routing congestion in practice (i.e., at finite blocklengths and non-vanishing error-probabilities), it has no impact on asymptotic bounds on total power. As proved in [12, Pages 36-37], for a process with L routing layers, the area occupied by wires is at least $\frac{A_{\text{wires}}}{L^2}$, where A_{wires} is the area occupied by the same circuit when only one metal layer is used. As long as L cannot grow with the number of vertices in the graph (it would be very unrealistic to assume it can), it has only a constant impact on wiring area lower bounds (see Lemmas 5, 6) and *no* impact (since one can always restrict routing to a single layer) on upper bounds (see Lemma 7). It will become apparent later in the paper therefore, that multiple routing layers do not effect any of the theoretical results we derive.

On the other hand, having multiple *active* layers with fine-grained routing between layers *can* lead to asymptotic reductions in wiring area for some circuits [36]. However, as it relates to practice, this is far beyond the reach of any commercial foundry in existence today. Methods for designing and fabricating such circuits (which rely on emerging nanotechnologies and emerging non-volatile memories [37]) are only now starting to be considered in research settings.

2) Architectural optimizations: Fully parallel, one clock-cycle per-iteration decoders are not commonly used in practice. Instead, serialization by dividing the number of physical nodes in the circuit by a constant factor and using time-multiplexing to cut down on wiring is often performed [4], [8]. This also requires a corresponding multiplication for the clock-frequency f_{clock} of the circuit to maintain the same data-rate. Recall, that dynamic power consumed in wires is proportional to $C_{\text{wires}} V_{supply}^2 f_{\text{clock}}$ [19]. While decrease in wire capacitance may allow the supply to be scaled down (leading to a reduction in power) without compromising timing, it is *not possible* to scale it down indefinitely, since transistors have a nonzero subthreshold slope [38, Section 2]. In other words, once a lower limit on supply voltage is reached, even if C_{wires} can be made to decay on the order of $\frac{1}{n}$, one would no longer achieve power savings due to the corresponding increase in f_{clock}. Thus, behavior of total power in the large blocklength limit will remain unchanged. Such architectural optimizations do however, have a big impact in practice (e.g., at finite-blocklengths) since changes in constants matter then.

3) Leakage power: Later in the paper (Sections IV-C, V-C), we will compare bounds on total power under the Node and Wire Models. It will turn out that the two models lead to very different insights, and the Wire Model results appear far more pessimistic. Which model then is closer to reality? It turns out that the Node Model is actually *very optimistic*. It assumes that each node consumes only constant energy per-iteration, *irrespective* of the clock period. From a circuit perspective, this is equivalent to assuming that the power consumption inside nodes is entirely dynamic [19], as the energy per-iteration does not increase with the clock-period. This is *far* from the reality in modern VLSI technologies. Transistors are not perfect switches [39], and *every* check-node and variable-node will consume a constant amount of *leakage power* while the decoder is on, *regardless* of clock period and switching activity. It is easy to see then, that even if the transistor leakage is very small, the decoding power *must* scale as $\Omega(n)$. For instance, even if the architecture is highly serialized, there is still leakage in each of the $\Theta(n)$ sequential elements (e.g., flip-flops, latches, or RAM cells) needed to store messages. It will become apparent later in the paper that this simple analysis is enough to establish identical conclusions to the lower bounds of Theorems 3, 4, and 5. Thus, the asymptotics of total power under the Wire Model should be viewed as *much* better predictions of what would actually happen inside the circuit at infinite blocklengths.

III. Preliminary results

In this section, we provide some preliminary results that will be useful in Sections IV and V. These include general bounds on the blocklength of regular-LDPC codes and bounds on the minimum number of independent iterations needed for Gallager decoders to achieve a specific bit-error probability.

A. Blocklength analysis of regular-LDPC codes

Lemma 1. *For a given girth g of a (d_v, d_c)-regular LDPC code, a lower bound on the blocklength n is*

$$n \geq \left[(d_v - 1)(d_c - 1)\right]^{\lfloor \frac{g-2}{4} \rfloor}, \tag{3}$$

and an upper bound on the blocklength is given by

$$n \leq 2(d_v + d_c) d_v d_c (2 d_v d_c + 1)^{\frac{3}{4}g}. \tag{4}$$

[7] Switching consumes energy because wires act as capacitors that need to be charged/discharged. If voltage is maintained, little additional energy is spent.

Proof: For the lower bound, see [14, Appendix I], and for the upper bound, see [14, Claim 2]. □

Lemma 2. *For a (d_v, d_c)-regular binary LDPC code decoded using any iterative message-passing decoding algorithm for any number of iterations, the blocklength n needed to achieve bit-error probability P_e is*

$$n = \begin{cases} \Omega\left(\left(\frac{\left(\frac{d_v-2}{d_v(d_v-1)}\right)^2 \log\frac{1}{P_e}}{(1+9\pi)\eta P_T}\right)^{\frac{1+\frac{\log(d_c-1)}{\log(d_v-1)}}{2}}\right) & d_v \geq 3. \\ \Omega\left(\left(\frac{1}{P_e}\right)^{\frac{1}{\eta P_T(1+9\pi)\left(2+\frac{2}{\log(d_c-1)}\right)}}\right) & d_v = 2. \end{cases}$$

Here, $\eta > 0$ is the constant attenuation in the AWGN channel (see Section II-A).

Proof: See Appendix A. □

PROOF OUTLINE: We use a technique for the finite-length analysis of LDPC codes from [40]. First, the *pairwise* error-probability for any iterative message-passing decoder is lower bounded in terms of n, d_v, d_c, and ηP_T using an expression for the minimum *pseudoweight* (see Appendix A for definition) of the code. Next, due to a simple relationship between bit-error probability and pairwise error-probability for binary linear codes over memoryless binary-input, output-symmetric channels, the bit-error probability can be lower bounded in terms of n, d_v, d_c, and ηP_T. Finally, algebraic manipulations, an application of (1), and an application of Definition 3 complete the proof. □

B. Approximation analysis of Gallager decoding algorithms

In this section, we bound the number of independent decoding iterations required to attain a specific bit-error probability with Gallager decoders. These bounds are used in Sections IV, V-B to prove achievability results for total power.

Lemma 3. *The number of independent decoding iterations N_{iter} needed to attain bit-error probability P_e with Gallager-A decoding is*

$$N_{\text{iter}} = \begin{cases} \Theta\left(\log\frac{1}{P_e}\right) & \text{if } P_T \text{ is held constant.} \\ \Theta\left(\frac{\log\frac{1}{P_e}}{\eta P_T}\right) & \text{if } P_T \text{ is not held fixed.} \end{cases}$$

Here, $\eta > 0$ is the constant attenuation in the AWGN channel (see Section II-A).

Proof: See Appendix B. □

PROOF OUTLINE: We first define (based on the decoding threshold over the BSC [25]) appropriate right-sided sets for analyzing the asymptotics of N_{iter} as a function of $\frac{1}{P_e}$ and P_T. Then, we apply a first-order Taylor expansion to the recurrence relation for bit-error probability under independent iterations of Gallager-A decoding from [25, Eqn. (6)] and carefully bound the approximation error. We then show that for small enough P_e or large enough P_T, the approximation error can be bounded by a multiplicative factor between $\frac{1}{2}$ and 1. After some algebraic manipulations and an application of (1), we apply Definition 2 to establish the first case and Definition 3 to establish the second case. □

Lemma 4. *The number of independent decoding iterations N_{iter} needed to attain bit-error probability P_e with a Gallager-B decoder with variable node degree $d_v \geq 4$ is given by*

$$N_{\text{iter}} = \begin{cases} \Theta\left(\log\log\frac{1}{P_e}\right) & \text{if } P_T \text{ is held constant.} \\ \Theta\left(\frac{\log\frac{\log\frac{1}{P_e}}{\eta P_T}}{\log\frac{d_v-1}{2}}\right) & \text{if } \lim_{P_e\to 0}\frac{P_T}{\log\frac{1}{P_e}} = 0. \end{cases}$$

Here, $\eta > 0$ is the constant attenuation in the AWGN channel (see Section II-A).

Proof: See Appendix C. Importantly, this holds only if $d_v \geq 4$, otherwise Gallager-A and Gallager-B are equivalent. Note that in the second case, we assume P_T is a function of $\frac{1}{P_e}$, so *both* expressions should be interpreted with Definition 2. Further, little generality is lost by the necessary condition for the second case, since uncoded transmission requires transmit power $\Theta\left(\log\frac{1}{P_e}\right)$ (see (1)). □

PROOF OUTLINE: We follow exactly the same steps as the proof of Lemma 3, but instead use a higher-order Taylor expansion of the recurrence relation for bit-error probability under Gallager-B decoding from [29, Eqn. 4.15]. □

IV. Analysis of Energy Consumption in the Node Model

In this section, we investigate the question: as $P_e \to 0$, how does the total power under the Node Model (see Section II-E) scale when Gallager decoders (restricted to independent iterations) are used?

A. Total power analysis for Gallager-A decoding

Corollary 1. *The optimal total power under Gallager-A decoding (restricted to independent iterations) in the Node Model (ξ_{node}) for a binary (d_v, d_c)-regular LDPC code is*

$$P_{\text{total,min}} = \Theta\left(\sqrt{\log\frac{1}{P_e}}\right)$$

which is achieved by transmit power $P_T^* = \Theta\left(\sqrt{\log\frac{1}{P_e}}\right)$.

Proof: Applying Lemma 3 to the Node Model, if P_T is held constant even as $P_e \to 0$, the power consumed by decoding is $\Theta\left(\log\frac{1}{P_e}\right)$. Since P_T is constant, the total power is also $P_{\text{total,bdd P}_\text{T}} = \Theta\left(\log\frac{1}{P_e}\right)$. If instead P_T is allowed to grow arbitrarily, the total power is given by

$$P_{\text{total}} = P_T + P_{\text{Dec}} = \Theta\left(P_T + \frac{\log\frac{1}{P_e}}{\eta P_T}\right). \tag{5}$$

Thus, optimizing the scaling behavior of the total power over

transmit power functions P_T

$$P_{\text{total,min}} = \min_{P_T} \Theta\left(P_T + \frac{\log \frac{1}{P_e}}{\eta P_T}\right) = \Theta\left(\sqrt{\log \frac{1}{P_e}}\right), \quad (6)$$

with optimizing transmit power $P_T^* = \Theta\left(\sqrt{\log \frac{1}{P_e}}\right)$. □

B. Total power analysis for Gallager-B decoding

Corollary 2. *The optimal total power under Gallager-B decoding (restricted to independent iterations) in the Node Model (ξ_{node}) for a binary (d_v, d_c)-regular LDPC code is*

$$P_{\text{total,min}} = \Theta\left(\log\log \frac{1}{P_e}\right),$$

which is achieved by transmit power $P_T^* = \Theta(1)$.

Proof: If P_T satisfies the condition stated in the second case of Lemma 4, the total power in the Node Model is

$$P_{\text{total}} = P_T + P_{\text{Dec}} = \Theta\left(P_T + \frac{\log \frac{\log \frac{1}{P_e}}{\eta P_T}}{\log \frac{d_v - 1}{2}}\right). \quad (7)$$

Minimizing the scaling behavior of (7), the optimizing transmit power is $P_T^* = \Theta(1)$. The optimal total power is then

$$P_{\text{total,min}} = \Theta\left(\log\log \frac{1}{P_e}\right). \quad (8)$$

In this case the optimizing transmit power is bounded even as $P_e \to 0$. □

C. Comparison with fundamental limits

Can we reduce the asymptotic growth of total power under the Node Model via a better code or a more sophisticated decoding algorithm? After all, we limited our attention to regular LDPCs and simple one-bit message-passing algorithms. It was shown in [6] that under the Node Model and a fully-parallelized decoding implementation such as Implementation Model (λ), the optimal total power is lower bounded by $\Omega\left(\log\log \frac{1}{P_e}\right)$, matching Corollary 2. In fact, using a code which performs close to Shannon capacity can even reduce efficiency for this strategy: if a capacity-approaching LDPC code is used instead of a regular LDPC code, the infinite-blocklength performance under the Gallager-B decoding algorithm equals that of regular-LDPCs with Gallager-A decoding. In other words, the bit-error probability decays only exponentially (and not doubly-exponentially) with the number of iterations under Gallager-B decoding if degree-2 variable nodes are present [41], and [42] shows that degree-2 variable nodes are *required* in order to achieve capacity (the fraction of degree-2 variable nodes required to attain capacity is characterized in [42]). Thus, rather than searching for an irregular code that approaches capacity, an engineer might be better off using a simpler regular code that approaches fundamental limits on total power.

V. Analysis of energy consumption in the wire model

A. Bounds on wiring area of decoders

To make use of the energy model of Section II-F, we must characterize the total wiring area of the decoder. We rely on techniques for upper and lower bounds on the total wire area obtained for different computations in [12], [43], [44], [45]. We first introduce some graph-theoretic concepts that will prove useful in obtaining similar bounds for our problem.

1) Lower bound on wiring area: We first provide a trivial lower bound on the wiring area of the decoder for any regular-LDPC code implemented in Implementation Model (λ).

Lemma 5. *For a (d_v, d_c)-regular LDPC code of blocklength n, the wiring area A_{wires} under Implementation Model (λ) is*

$$A_{\text{wires}} \geq \lambda^2 d_v n.$$

Proof: There are $d_v n$ wires. Each wire has width λ and minimum length λ (no two wires overlap completely). □

In his thesis [45], Leighton utilizes the *crossing number* (a property first defined by Turán [46]) of a graph as a tool for obtaining lower bounds on the wiring area of circuits. Crossing numbers continue to be of interest to combinatorialists and graph-theorists, and many difficult problems on finding exact crossing numbers or bounds for various families of graphs remain open [47]. We use the following two definitions to introduce this property.

Definition 8 (Graph Drawing)**.** A drawing of a graph $\mathcal{G}$ is a representation of $\mathcal{G}$ in the plane such that each vertex of $\mathcal{G}$ is represented by a distinct point and each edge is represented by a distinct continuous arc connecting the corresponding points, which does not cross itself. No edge passes through vertices other than its endpoints and no two edges are overlapping for any nonzero length (they can only intersect at points).

Definition 9 (Crossing Number)**.** The crossing number of a graph $\mathcal{G}$, $cr(\mathcal{G})$, is the minimum number of edge-crossings over all possible drawings of $\mathcal{G}$. An edge-crossing is any point in the plane other than a vertex of $\mathcal{G}$ where a pair of edges intersects.

For any graph $\mathcal{G}$ (e.g., the Tanner graph of an LDPC code), the wiring area of the corresponding circuit under Implementation Model (λ) is lower bounded as $A_{\text{wires}} \geq \lambda^2 cr(\mathcal{G})$. This is due to the fact that any VLSI layout of the type described in Section II-C can be mapped to a drawing of $\mathcal{G}$ in the sense of Definition 8, by simply replacing each processing node with a point in the plane and replacing each wire by line segments connecting two points. Therefore, the minimum number of wire crossings of any layout of $\mathcal{G}$ is $cr(\mathcal{G})$. Since every crossing has area λ^2, the inequality follows. We now need lower bounds on the crossing number of a computation graph. In this paper, we make use of the following result [48] that improves on earlier results [49], [50], [45] and allows us to tighten Lemma 5 for some codes.

Theorem 1 (Pach, Spencer, Tóth [48])**.** *Let $\mathcal{G} = \{V, E\}$ be a graph with girth $g > 2\ell$ and $|E| \geq 4|V|$. Then $cr(\mathcal{G})$ satisfies*

$$cr(\mathcal{G}) \geq k_\ell \frac{|E|^{\ell+2}}{|V|^{\ell+1}},$$

where $k_\ell = \Omega\left(\frac{1}{\ell^2 2^{2\ell+3}}\right)$ [51].

We now obtain lower bounds on wiring area given a lower

bound on the number of independent iterations the code allows.

Lemma 6 (Crossing Number Lower Bound on A_{wires})**.** *For a (d_v, d_c)-regular LDPC code that allows for at least $\underline{N}_{\text{iter}}$ independent decoding iterations, the wiring area A_{wires} of a decoder in Implementation Model (λ) is*

$$A_{\text{wires}} = \begin{cases} \Omega\left(e^{\gamma \underline{N}_{\text{iter}}}\right) & \text{for any } d_v,\ d_c \\ \Omega\left(e^{\underline{N}_{\text{iter}} \log \frac{2d_v^2 d_c^2}{(d_v+d_c)^2}}\right) & \text{if } d_v d_c \geq 4(d_v+d_c). \end{cases}$$

Here, $\gamma \in [\log[(d_v-1)(d_c-1)], 3\log(2d_v d_c+1)]$ is a constant that depends on the code construction.

Proof: Let $\mathcal{C}$ be a (d_v, d_c)-regular LDPC code that allows for *at least* $\underline{N}_{\text{iter}}$ independent decoding iterations. Since the girth g of $\mathcal{C}$ must then satisfy $\lfloor \frac{g-2}{4} \rfloor \geq \underline{N}_{\text{iter}}$, $g > 4\underline{N}_{\text{iter}} - 2$. From Lemma 1 then, the blocklength n of the code $\mathcal{C}$ is

$$n = \Omega\left(e^{\gamma \underline{N}_{\text{iter}}}\right),$$

where $\gamma \in [\log((d_v-1)(d_c-1)), 3\log(2d_v d_c+1)]$. And from Lemma 5 we then have $A_{\text{wires}} = \Omega\left(e^{\gamma \underline{N}_{\text{iter}}}\right)$. Now, assume $d_v d_c \geq 4(d_v+d_c)$. This requires that $d_c > d_v \geq 5$. Let $V_{\mathcal{C}}$, $E_{\mathcal{C}}$ denote the sets of vertices and edges in the Tanner graph of $\mathcal{C}$. The sizes are $|E_{\mathcal{C}}| = nd_v$ and $|V_{\mathcal{C}}| = n\left(1+\frac{d_v}{d_c}\right)$. We then carry out the following algebra

$$d_v d_c \geq 4(d_v+d_c) \Rightarrow nd_v \geq 4n\left(1+\frac{d_v}{d_c}\right).$$

Hence, $|E_{\mathcal{C}}| \geq 4|V_{\mathcal{C}}|$. Using the fact that $g > 4\underline{N}_{\text{iter}} - 2$, we apply Theorem 1

$$\begin{aligned} A_{\text{wires}} &= \Omega\left(\frac{\lambda^2}{(2\underline{N}_{\text{iter}}-1)^2 4^{2\underline{N}_{\text{iter}}+\frac{1}{2}}} \frac{(nd_v)^{2\underline{N}_{\text{iter}}+1}}{\left(n\left(1+\frac{d_v}{d_c}\right)\right)^{2\underline{N}_{\text{iter}}}}\right) \\ &= \Omega\left(\frac{\lambda^2\left(\frac{e^\gamma}{16}\right)^{\underline{N}_{\text{iter}}}}{(2\underline{N}_{\text{iter}}-1)^2}\left(\frac{d_v d_c}{d_v+d_c}\right)^{2\underline{N}_{\text{iter}}}\right). \end{aligned} \tag{9}$$

Then, because $e^\gamma \geq (d_v-1)(d_c-1) = d_v d_c - (d_v+d_c) + 1 \overset{d_v d_c \geq 4(d_v+d_c)}{\geq} 3(d_v+d_c)+1$, and because $d_c > d_v \geq 5$, we must have $e^\gamma \geq 34$. Substituting into (9),

$$\begin{aligned} A_{\text{wires}} &= \Omega\left(\frac{\lambda^2\left(\frac{34}{16}\right)^{\underline{N}_{\text{iter}}}}{(2\underline{N}_{\text{iter}}-1)^2}\left(\frac{d_v d_c}{d_v+d_c}\right)^{2\underline{N}_{\text{iter}}}\right) \\ &= \Omega\left(\lambda^2 2^{\underline{N}_{\text{iter}}}\left(\frac{d_v d_c}{d_v+d_c}\right)^{2\underline{N}_{\text{iter}}}\right), \end{aligned}$$

and changes-of-base complete the proof. □

2) Upper bound on wiring area: Since the total circuit area is always an upper bound on the area occupied by wires, we use an upper bound on the circuit area to obtain the following upper bound on the wiring area based on the *maximum* number of independent iterations that the code allows for.

Lemma 7 (Upper bound on A_{wires})**.** *For a (d_v, d_c)-regular LDPC code that allows for no more than $\overline{N}_{\text{iter}}$ independent decoding iterations, the decoder wiring area A_{wires} is*

$$A_{\text{wires}} = \mathcal{O}\left(e^{2\gamma \overline{N}_{\text{iter}}}\right).$$

Here, $\gamma \in [\log((d_v-1)(d_c-1)), 3\log(2d_v d_c+1)]$ is a constant that depends on the code construction.

Proof: Let $\mathcal{C}$ be a (d_v, d_c)-regular LDPC code that allows for no more than $\overline{N}_{\text{iter}}$ independent decoding iterations. Since the girth g of $\mathcal{C}$ must then satisfy $\lfloor \frac{g-2}{4} \rfloor \leq \overline{N}_{\text{iter}}$,

$$g < 4\overline{N}_{\text{iter}} + 6.$$

From Lemma 1, the blocklength of any such code can be upper bounded in the order of $\overline{N}_{\text{iter}}$ as

$$n = \mathcal{O}\left(e^{\gamma \overline{N}_{\text{iter}}}\right), \tag{10}$$

where $\gamma \in [\log((d_v-1)(d_c-1)), 3\log(2d_v d_c+1)]$. Then, consider a "collinear" VLSI layout [52] of the Tanner graph of $\mathcal{C}$ which satisfies all the assumptions described in Section II-C. Arrange all variable-nodes and check-nodes in the graph along a horizontal line, leaving λ spacing between consecutive nodes. The total length of this arrangement is then $\mathcal{O}(n)$. Allocate a unique horizontal wiring track for each of the nd_v edges in the Tanner graph. Then, every connection in the graph can be made with two vertical wires (one from each endpoint) which connect to the opposite ends of the dedicated horizontal track. The total height of this layout is then $\mathcal{O}(n)$, and the total area is $\mathcal{O}(n^2)$. An example collinear layout is given in Fig. 3. Substituting (10) for n, we obtain the bound. □

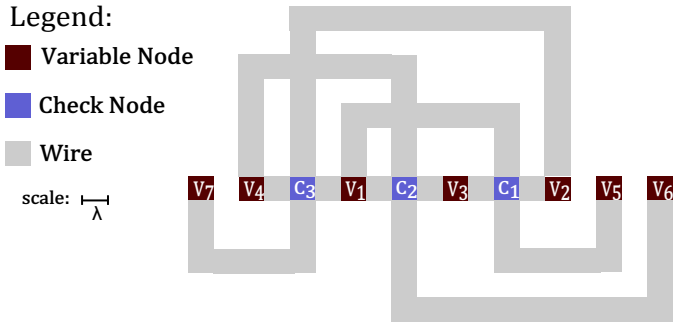


Fig. 3: An example collinear layout for the same $(7,4)$ Hamming Code depicted in Fig. 2.

We note that this upper bound is crude since the $\mathcal{O}\left(|V|^2\right)$ layout construction applies for any graph $\mathcal{G} = \{V, E\}$ which satisfies $|E| = \mathcal{O}(|V|)$. A simple proof [45] shows that one can create a layout of area $\mathcal{O}\left((|V| + cr(\mathcal{G}))\log(|V| + cr(\mathcal{G}))\right)$ for any graph. Thus, an algorithm for drawing semi-regular graphs which can be proven to yield sub-quadratic (in n) crossing numbers would yield energy-efficient codes and decoders with short wires.

B. Total power minimization for the wire model

We now present analogues of results in Section IV, where we instead consider decoding power described by the Wire Model of Section II-F. We translate the wiring area bounds of Section V-A to power bounds.

Theorem 2 (Asymptotic bounds on P_{wires})**.** *Under Implementation Model (λ) and Wire Model (ξ_{wire}), the decoding power P_{wires} for a (d_v, d_c)-regular binary LDPC code that allows for* exactly N_{iter} *independent iterations is bounded as*

$$P_{\text{wires}} = \begin{cases} \Omega\left(e^{\gamma N_{\text{iter}}}\right) & \text{for any } d_v,\ d_c \\ \Omega\left(e^{N_{\text{iter}} \log \frac{2d_v^2 d_c^2}{(d_v+d_c)^2}}\right) & \text{if } d_v d_c \geq 4(d_v+d_c) \\ \mathcal{O}\left(e^{2\gamma N_{\text{iter}}}\right) & \text{for any } d_v,\ d_c \end{cases}$$

where $\gamma \in [\log((d_v-1)(d_c-1)), 3\log(2d_v d_c+1)]$ is a constant that depends on the code construction..

Proof: The result is a straightforward conclusion from Lemma 6 and Lemma 7 applied in Definition 7. □

Next, we present a general lower bound on the scaling behavior of total power under the Wire Model for any binary regular-LDPC code, decoded using any iterative message-passing decoding algorithm, for any number of iterations.

Theorem 3 (Lower bound for regular-LDPCs)**.** *The optimal total power in the Wire Model (ξ_{wire}) for a binary (d_v, d_c)-regular LDPC code with any iterative message-passing decoding algorithm to achieve bit-error probability P_e is*

$$P_{\text{total,min}} = \Omega\left(\log^{\frac{1}{1+\frac{2}{1+\frac{\log(d_c-1)}{\log(d_v-1)}}}} \frac{1}{P_e}\right).$$

Further, if P_T is held fixed as $P_e \to 0$ the total power diverges as $\Omega\left(\log^y \frac{1}{P_e}\right)$ where $y > 1$, which dominates the power required by uncoded transmission.

Proof: See Appendix D. □

PROOF OUTLINE: We first substitute the result of Lemma 2 into Lemma 5, and then use the resulting lower bound on decoding power under the Wire Model in (2). Using simple calculus, we then derive the asymptotics of the transmit power function that minimizes the total power, and plug it back into Lemma 2 and (2) to obtain the result. □

1) Gallager-A decoding:

Theorem 4. *The optimal total power under Gallager-A decoding (restricted to independent iterations) in the Wire Model (ξ_{wire}) for a binary (d_v, d_c)-regular LDPC code to achieve bit-error probability P_e is*

$$P_{\text{total,min}} = \Theta\left(\frac{\frac{\gamma}{\eta}\log\frac{1}{P_e}}{\log\log\frac{1}{P_e}}\right),$$

Where $\eta > 0$ is the constant attenuation in the AWGN channel (Section II-A) and $\gamma \in [\log((d_v-1)(d_c-1)), 3\log(2d_v d_c+1)]$ is a constant that depends on the code construction. Further, if P_T is held fixed as $P_e \to 0$, then total power diverges as $\Omega\left(Poly\left(\frac{1}{P_e}\right)\right)$, which is an exponential function of the power required by uncoded transmission.

Proof: See Appendix E. □

PROOF OUTLINE: We first substitute the results of Lemma 3 into Theorem 2, and then plug in the resulting bounds on decoding power in (2). Using some calculus, we then derive the best-case and worst-case asymptotics of the transmit power function that minimizes the total power, and show that there is at most a constant gap between the two. We then plug the optimizing transmit power back into Lemma 3 and (2) to obtain the result. □

2) Gallager-B decoding:

Theorem 5. *The optimal total power under Gallager-B decoding (restricted to independent iterations) in the Wire Model (ξ_{wire}) for a binary (d_v, d_c)-regular LDPC code to achieve bit-error probability P_e is bounded as*

$$P_{\text{total,min}} = \begin{cases} \Omega\left(\log^{\frac{2}{3}} \frac{1}{P_e}\right) & \frac{d_v d_c}{(d_v+d_c)} < 4 \\ \Omega\left(\log^{\frac{31}{40}} \frac{1}{P_e}\right) & \frac{d_v d_c}{(d_v+d_c)} \geq 4 \\ \mathcal{O}\left(\log^{\frac{1}{1+\frac{\log(d_v-1)-\log 2}{6\log(2d_v d_c+1)}}} \frac{1}{P_e}\right) & \text{any } d_v,\ d_c \end{cases}$$

Further, if P_T is held fixed as $P_e \to 0$, then total power diverges as $\Omega\left(\log^{2.48} \frac{1}{P_e}\right)$, which is a super-quadratic function of the power required by uncoded transmission.

Proof: See Appendix F. □

PROOF OUTLINE: We first substitute the results of Lemma 4 into Theorem 2, and then plug in the resulting bounds on decoding power in (2). We then use algebraic manipulations to bound the exponents in Theorem 2. Next, we use calculus to derive the best-case and worst-case asymptotics of the transmit power function that minimizes the total power. We then plug the optimizing transmit power into Lemma 4 and (2) to obtain the results. □

C. Comparison with fundamental limits

In [13], using a more pessimistic Wire Model[8], it is shown that the total power required for any error-correcting code and any message-passing decoding algorithm is fundamentally lower bounded by $\Omega\left(\log^{\frac{1}{3}} \frac{1}{P_e^{blk}}\right)$, where P_e^{blk} is the *block*-error probability. Theorem 3 shows that regular-LDPC codes with iterative message-passing decoders cannot do better than $\Omega\left(\log^{\frac{1}{2}} \frac{1}{P_e}\right)$ where P_e is bit-error probability, and the exponent $\frac{1}{2}$ can only be obtained in the limit of large degrees and vanishing code-rate. Since block-error probability exceeds bit-error probability, regular-LDPC codes do not achieve fundamental limits[9] on total-power in the Wire Model.

Theorem 4 is the first constructive result that shows that coding can (asymptotically) outperform uncoded transmission in total power for the Wire Model. However, the gap in total power between the two is merely a multiplicative factor of $\log\log\frac{1}{P_e}$. While Theorem 5 proves that it is possible to increase the relative advantage of coding to a fractional power of $\log\frac{1}{P_e}$, the difference between the upper bound and the power for uncoded transmission is minuscule. The exponent of $\log\frac{1}{P_e}$ in the upper bound is an increasing function of both

[8]The Wire Model of [13] assumes the power is proportional to $A_{\text{wires}} N_{\text{iter}}$. Here it is assumed to be simply proportional to A_{wires}.

[9]Though, this may simply mean the fundamental limits [13] are not tight.

d_v and d_c, approaching 1 as either gets large. Since Gallager-B decoding requires $d_v \geq 4$, the smallest exponent for regular LDPCs occurs when $d_v = 4$ and $d_c = 5$. The numerical value of the exponent for these degrees is ≈ 0.98, which suggests little order sense improvement over uncoded transmission. Hence, the wiring area at the decoder (particularly, how much better it can be than the bound of Lemma 7) is crucial in determining how much can be gained by using Gallager-B decoding instead of uncoded transmission. Further discussion is provided in Section VII.

VI. Circuit simulation based numerical results

At reasonable bit-error probabilities (e.g., 10^{-5}) and short distances (e.g., less than five meters), asymptotic bounds cannot provide precise answers on which codes to use. For example, consider the following problem, shown graphically in Fig. 1b).

Problem 1. Suppose we want to design a point-to-point communication system that operates over a given channel. We are given a target bit-error probability P_e, communication distance r, and system data-rate R_{data} that the link must operate at. Which code and corresponding decoding algorithm minimize the total (i.e. transmit + decoding) power?

Since the bounds of Sections II-V are derived as $P_e \to 0$, they may not be applicable to many instances of Problem 1. In this section we therefore develop a methodology for rapidly exploring a space of codes and decoding algorithms to answer specific instances of Problem 1. We focus on one-bit Gallager A and B [29] and two-bit [53] decoding algorithms, restricting the number of algorithmic iterations to $\lfloor \frac{g-2}{4} \rfloor$. Because of the effort required in implementing or even simulating a single decoder in hardware, we construct models[10] for power consumed in decoding implementations of different algorithms based on post-layout circuit simulations for simple check-node and variable-node circuits. The models developed attempt to capture detailed *physical* aspects (e.g., interconnect lengths and impedance parameters, propagation delays, silicon area, and power-performance tradeoffs) of implementations, in stark contrast with their theoretical counterparts of Sections II-V. In Section VI-C, we use these models to investigate solutions to some instances of Problem 1.

A. Note on channels and constellation size

To answer Problem 1, additional physical assumptions about the channel (e.g., bandwidth, fading, path-loss, temperature, constellation size) are required in comparison to the model of Section II-A. The channel is still assumed to be AWGN with fixed attenuation. However, while Section II-A assumes BPSK modulation for all transmissions, due to the introduction of a data-rate constraint and fixed passband bandwidth W (for fair comparison), the constellation size is required to vary based on the code rate. Explicitly, the transmission strategy is assumed to use either BPSK or square-QAM modulation, mapping codeword bits to constellation symbols. We assume that if square-QAM modulation is used, the information bits are mapped onto the constellation signals using a two-dimensional Gray code as explained in [56, Section III]. We assume the transmitter signals at a rate of W symbols/s and that the minimum square constellation size (M) satisfying the system data-rate requirement is chosen: M is always the smallest square of an even integer for which:

$$M \geq 2^{R_{\text{data}}/(W \times R)}.$$

For calculating transmit power numbers, the thermal noise variance used is $\sigma_z^2 = kTW$, where k is the Boltzmann constant (1.38×10^{-23} J/K), and T is the temperature. The power is assumed to decay according to a power-law path-loss model $1/r^{\alpha}$, where α is the path-loss coefficient. the received $\frac{E_b}{N_0}$ is obtained as a function of the system and channel parameters:

$$\frac{E_b}{N_0} = \frac{P_T}{kTW\left(\frac{r}{\lambda}\right)^{\alpha}\log_2(M)}, \tag{11}$$

where λ is the wavelength of transmission at center frequency f_c in Hz ($\lambda = 3 \times 10^8/f_c$). The channel flip probability for BPSK transmissions under this model is $p_0 = \mathbb{Q}\left(\sqrt{\frac{2E_b}{N_0}}\right)$, and the channel flip probability for M-ary square QAM is [56, Section III.B]:

$$p_0 = \frac{1}{\log_2(\sqrt{M})} \sum_{k=1}^{\log_2(\sqrt{M})\left(1-\frac{1}{2^k}\right)\sqrt{M}-1} \sum_{j=0} \left[(-1)^{\left\lfloor \frac{j\times 2^{k-1}}{\sqrt{M}} \right\rfloor} \times \left(2^{k-1} - \left\lfloor \frac{j \times 2^{k-1}}{\sqrt{M}} + \frac{1}{2} \right\rfloor\right) \times 2\mathbb{Q}\left((2j+1)\sqrt{\frac{3\frac{Eb}{N_0}\log_2(M)}{(M-1)}}\right)\right]. \tag{12}$$

Also, note that the asymptotic bounds derived in Sections II-V *remain unchanged*, even if we substitute M-ary QAM for BPSK as the signaling constellation. This follows from the fact that the RHS of equation (12) is a linear combination of $\mathbb{Q}(\cdot)$ functions with argument linearly proportional to $\sqrt{\frac{E_b}{N_0}}$. Hence, even for M-ary QAM, $p_0 = \Theta\left(\frac{e^{-\phi P_T}}{\sqrt{4\pi\phi P_T}}\right)$ for some constant $\phi \neq \eta$ (see (1)). Since the difference is merely a constant, the asymptotic analysis of Sections II-V holds.

For the results presented in Section VI-C, we assume the decoding throughput is required to be equal to $R_{\text{data}} = 7$ Gb/s. We assume a channel center frequency of $f_c = 60$ GHz and bandwidth of $W = 7$ GHz. The temperature T is 300 K. The distances considered are much larger than the wavelength of transmission (≈ 0.5 cm) so the "far-field approximation" applies.

B. Simulation-based models of LDPC decoders

Given a code, decoding algorithm, and desired data-rate, calculating the required decoding power is a difficult task. Even within the family of regular LDPC codes and specified decoding algorithms, the decoder can be implemented in myriad ways. The choice of circuit architecture, implementation technology, and even process-specific transistor options can

[10] These models have been created in a open-access CMOS library [54] and are online at [55].

have a significant impact on the decoding power [8], [4]. A *comprehensive solution* to Problem 1 requires optimization of total power over not just *super-exponentially* many codes and decoding algorithms, but also *all* decoder *architectures*, *implementation technologies*, and *process options*, which could be an impossibly hard problem. The models we present here are based on simulations of synchronous, fully-parallel decoding architectures in a 32/28nm CMOS process with a high threshold voltage, and are used in Section VI-C to obtain insights on the nature of optimal solutions. We believe that incorporating more models of this nature and performing the resulting optimization could be a good approach to obtain low total power solutions. We now describe how the model is generated.

1) Initial post-layout simulations: Our models for arbitrary-blocklength LDPC decoders are constructed based on circuit simulations using the Synopsys 32/28nm high threshold voltage CMOS process with 9 metal-layers [54]. First, post-layout simulations of check-node and variable-node circuits for one-bit and two-bit decoders are performed. The physical area, power consumption, and critical-path delays of the check-nodes and variable-nodes are used as the basis for our models. The CAD flow used is detailed in Appendix G. The next section details how these results are generalized to full decoders.

2) Physical model of LDPC decoding: Even within our imposed restrictions on the LDPC code degrees, girth, and number of message-passing bits for decoding, constructing a decoding power model that applies to all combinations of these code parameters requires some assumptions:

1. Decoders operate at a fixed supply voltage (chosen as 0.78V: the minimum supply voltage of the timing libraries included with the standard-cell library).
2. The code design space includes regular-LDPC codes with variable-node degrees $2 \leq d_v \leq 6$, check-node degrees $3 \leq d_c \leq 13$, and girths $6 \leq g \leq 10$.
3. "Minimum-Blocklength" codes (found in [57]) are chosen for a given g, d_v, d_c. Hence the blocklength is expressed as a function of these parameters: $n^{(\min)}_{g,d_v,d_c}$.
4. The decoding algorithm a, is chosen from the set $\{A, B, T\}$, where A, B, T correspond to Gallager-A, Gallager-B, and Two-bit[11] [53] message-passing decoding algorithms, respectively. We use $\#_{\text{bits}}(a)$ to refer to the number of message bits used in algorithm a.

We then model the minimum-achievable clock period T_{CLK}, and maximum-achievable decoding throughput R_{Dec} for each decoder as functions of a, g, d_v, d_c:

$$\begin{aligned} T_{\text{CLK}}(a,g,d_v,d_c) &= T_{\text{VN}}(a,d_v) \\ &+ 2T_{\text{wire}}(a,g,d_v,d_c) \\ &+ T_{\text{CN}}(a,d_c) \end{aligned} \quad (13)$$

$$R_{\text{Dec}}(a,g,d_v,d_c) = \frac{n^{(\min)}_{g,d_v,d_c}\left(1-\frac{d_v}{d_c}\right)}{\lfloor \frac{g-2}{4} \rfloor \times T_{\text{CLK}}(a,g,d_v,d_c)} \quad (14)$$

In (13), $T_{\text{VN}}(\cdot,\cdot)$ and $T_{\text{CN}}(\cdot,\cdot)$ are critical-path delays through variable and check nodes respectively and $T_{\text{wire}}(\cdot,\cdot,\cdot,\cdot)$ is the propagation delay through a single message-passing interconnect. In essence, (13) formulates the critical-path delay for the decoder by summing up the propagation delays of all logic stages traversed in a single decoding iteration. Details for each component are given in Appendix H. We model the decoding power as

$$\begin{aligned} P_{\text{Dec}}(a,g,d_v,d_c) &= n^{(\min)}_{g,d_v,d_c}\Bigg[P_{\text{VN}}(a,d_v) + \frac{d_v P_{\text{CN}}(a,d_c)}{d_c} \\ &+ 2d_v \#_{\text{bits}}(a) \times P_{\text{wire}}(a,g,d_v,d_c)\Bigg] \end{aligned} \quad (15)$$

In (15), $P_{\text{VN}}(\cdot,\cdot)$ and $P_{\text{CN}}(\cdot,\cdot)$ are the power consumed in individual variable and check nodes respectively, and $P_{\text{wire}}(\cdot,\cdot,\cdot,\cdot)$ is the power consumed in a single message-passing interconnect. Note that (15) is a sum of all power consumed in computations and wires of the decoder (the coefficients in (15) count the number of occurrences of each power sink in the decoder). The details of the node power models are given in Appendix I and the details of the wire power model are given in Appendix J.

3) Satisfying the communication data-rate: Fixing the supply voltage for a decoder and using the fastest possible clock speed only allows for a single decoding throughput. Hence, parallelism in order to meet the system data-rate requirement R_{data} in Problem 1 is also modeled. For example, two copies of a single decoder can be used in parallel. Together, they provide twice the throughput, and require twice the power of a single decoder. In the corresponding communication system architecture, two separate codewords are required to be transmitted at twice the throughput of a single decoder, and a multiplexer at the receiver must pass a separate codeword to each of the parallel decoders, which decode the two codewords independently. Though making such a design choice in practice would introduce additional hardware and a slight power consumption overhead, we ignore this cost in our analysis.

In cases where integer multiples of a single decoder's throughput do not exactly reach R_{data}, we first find the minimum number of parallel decoders, that when combined, exceed the required throughput. Calling this minimum number of decoders $\mathcal{Q}$, we then assume that the clock period of each of the parallel decoders is increased until the overall throughput of the parallel combination is exactly R_{data}. Explicitly, the formula to determine this "underclocked" period T_u is:

$$T_u = \frac{\mathcal{Q} \times n^{(\min)}_{g,d_v,d_c}\left(1-\frac{d_v}{d_c}\right)}{\lfloor \frac{g-2}{4} \rfloor \times R_{\text{data}}}. \quad (16)$$

Because the decoding power is modeled as inversely proportional to the decoder clock period (see Appendices I-J), we multiply each individual decoder's power by the appropriate scaling factor $\kappa = \frac{T_{\text{CLK}}(a,g,d_v,d_c)}{T_u}$, and then multiply the result by the number of parallel decoders to get the total power of the parallel combination:

$$P_{\text{parallel}} = \mathcal{Q} \times P_{\text{Dec}}(a,g,d_v,d_c) \times \kappa. \quad (17)$$

We substitute (14), (16), and carry out some algebra to obtain:

$$P_{\text{parallel}} = P_{\text{Dec}}(a,g,d_v,d_c) \times \frac{R_{\text{data}}}{R_{\text{Dec}}(a,g,d_v,d_c)}. \quad (18)$$

[11] With fixed decoding algorithm parameters chosen as $C = 2$, $S = 2$, $W = 1$, for reasons explained in [53, Section II].

Hence, we assume that any (throughput, power) pair that is a multiple of the specifications of the original decoder can be achieved in this manner (with the obvious exception of points that have negative throughput and power). Therefore, in our analysis in Section VI-C, we assume the decoding throughput is exactly R_{data} and we use the decoding power numbers obtained via this interpolation between the modeled points.

4) Comparing different coding strategies: Now, given a subset of codes and decoders, how should a system designer jointly choose a code and decoding algorithm to minimize the total system power? Within the channel model of Section VI-A, consider specific *instances* of Problem 1: let path-loss coefficient α and R_{data} be fixed. Then, for each choice of (r, P_e), we can compare the required total power for each combination of code and decoding algorithm modeled in Section VI-B2, and find the minimizing combination.

C. Example: 60 GHz point-to-point communication

An example plot which shows the minimum achievable total power for different P_e values at a fixed distance $r = 3.2$m and $\alpha = 3$ is given in Fig. 4. The plot also shows the curve of the optimizing transmit power, P_T^*, and the Shannon-limit [58] for the AWGN channel. The horizontal gap between the optimizing P_T curve and the total power curve in Fig. 4 corresponds to the optimizing decoding power. As P_e decreases, this gap increases, indicating an increase in the total power-minimizing decoder's complexity.

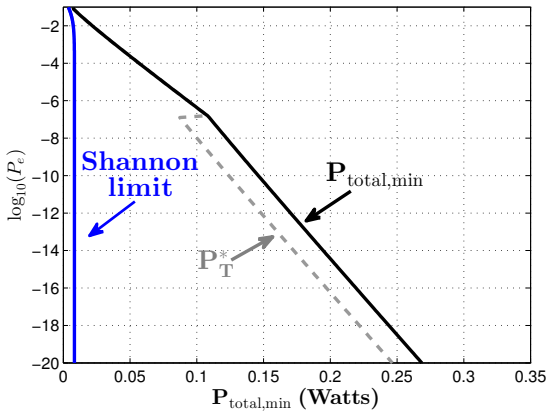


Fig. 4: A plot of $\log_{10}(P_e)$ vs. minimum achievable total power for $\alpha = 3$ at a fixed distance of $r = 3.2$m. The Shannon limit for the channel and the optimizing transmit power are also shown.

1) Joint optimization over code-decoder pairs: The form of the total power curve varies with communication distance. For improved understanding, we use two-dimensional contour plots in the (r, P_e) space to evaluate choices of codes and decoders, as suggested by Fig. 1b). An example is shown in Fig. 5, which compares code and decoding algorithm choices for path-loss coefficient $\alpha = 3$. In the top plot, the contours represent regions in the (r, P_e) space where specific combinations minimize total power, and in the bottom plot, regions in the (r, P_e) space are divided based on the value of the minimum total power. The best choices for these instances of Problem 1 turn out to be rate $\frac{1}{2}$ codes. Lower rate codes require large constellations for a 7 Gb/s data-rate, thus requiring large transmit power for the same p_0, and higher rate codes require larger decoding power due to increased complexity and size of higher degree nodes. Some tradeoffs between total power and code and decoder complexity can also be observed in Fig. 5: to minimize total power, algorithm complexity a should increase with r and code girth g should increase with decreasing P_e.

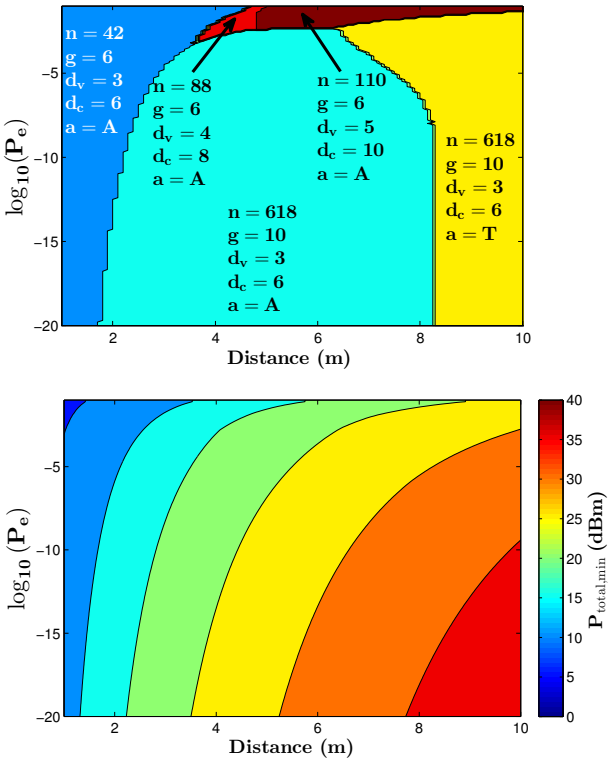


Fig. 5: Contour plots of the optimizing code & decoding algorithm choice (top) and the minimum total power in dBm (bottom). For these plots, $\alpha = 3$. The contours in the top plots are labeled with blocklength n, code girth g, VN degree d_v, CN degree d_c, and decoding algorithm a of the optimizing code and decoder. To interpret the plots, one can choose any point in the (r, P_e) space and find the best coding strategy (within the search space) in the top plot and the required total power to implement it in the bottom plot. The plot is best viewed in color.

How does the inclusion of uncoded transmission as a possible strategy change the picture? Contour plots with uncoded transmission included are given in Fig. 6. Comparing Fig. 6 with Fig. 5, we see that when uncoded transmission is included, it overtakes areas in the (r, P_e) space where P_e is high and r is very small. However, Fig. 6 suggests that simple codes and decoders can still outperform uncoded transmission at reasonably low P_e and distances of several meters or more.

VII. Conclusions and Discussions

In this work, we performed asymptotic analysis of the total (transmit + decoding) power for regular-LDPC codes with iterative-message passing decoders. While these codes (with Gallager-B decoding) can achieve fundamental limits in the Node Model [6], they are unable to do so for the Wire Model [13]. This suggests that measuring complexity of decoding by simply counting the number of operations

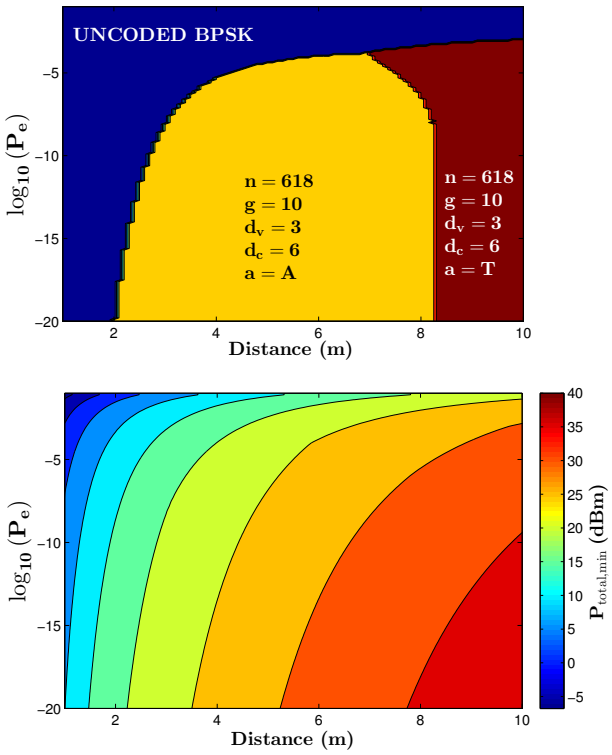


Fig. 6: Contour plots of the optimizing code & decoding algorithm choice (top) and the minimum total power in dBm (bottom), including uncoded transmission in the optimization space. For these plots, $\alpha = 3$. The contours in the top plots are labeled with blocklength n, code girth g, VN degree d_v, CN degree d_c, and decoding algorithm a of the optimizing code and decoder. To interpret the plots, one can choose any point in the $(r,\ P_e)$ space and find the best coding strategy (within the search space) in the top plot and the required total power to implement it in the bottom plot. The plot is best viewed in color.

(e.g., [59], [60], [61]) is insufficient for understanding system-level power consumption. In fact, for the Wire Model, even achieving order-sense advantage over *uncoded transmission* requires that both transmit and decoding power diverge to ∞ as $P_e \to 0$, which calls into question the assumption that one should fix the transmit power and operate near the Shannon capacity in order communicate reliably at a low power cost. However, this analysis also established a result of intellectual interest: that regular-LDPC codes *can* achieve an order-sense improvement in total power over uncoded transmission as bit-error probability tends to zero. This question only arises from the total power perspective adopted in this work, and it suggests that these results are only scratching the surface of a deeper theory in this direction.

To establish some constructive results, we analyzed two strategies where the number of decoding iterations is dictated by the girth of the code. Although this is convenient for proving asymptotic upper bounds on total power, this is rarely followed in practice. Typically, combinatorial properties of the code construction are analyzed and simulations are performed [30] in order to discern the error-probability behavior in decoding iterations beyond the girth-limit. However, we are not sure if better asymptotics for total power can be achieved by merely adding these additional iterations.

Our work highlights an important question that has received little attention in coding theory literature: design of codes that have good performance while maintaining small wiring area (see [62], [63], [64] for some heuristic approaches to generating Tanner graphs with low wiring complexity). For wire power consumption, there is a significant gap between the bounds on power consumed by regular-LDPC codes and iterative message-passing decoders derived here, and the fundamental limits derived in [13]. Nevertheless, even though regular-LDPC codes might not achieve these fundamental limits (and the fundamental limits themselves may not be tight), it is important to investigate wiring complexity of other coding families, such as Polar codes [61] and Turbo codes [65].

Recent work of Blake and Kschischang [18] studied the limiting bisection-width [12] of sequences of bipartite graphs with the size of the left-partite set tending to infinity, when the limiting degree-distributions of the left and right partite sets satisfy a certain sufficient condition [18, Theorem 1]. It is shown that when sequences satisfying this condition are generated by a standard uniform random configuration model (see [18, Section IV] for definition), the resulting graphs have a super-linear (in the number of vertices) bisection-width in the limit of the sequence with probability 1. In Corollary 2 and Section IV. A of [18], the authors show that the Tanner graphs of all capacity-approaching LDPC sequences as well as some regular-LDPC sequences generated using this method will satisfy the sufficient conditions. A super-linear bisection width for a graph implies that the area of wires in the corresponding VLSI circuit must scale at least quadratically in the number of vertices [12]. If using the decoding strategy of Theorem 5 then, such sequences of codes will have minimum total power that is $\Theta\left(\log^m \frac{1}{P_e}\right)$, where $0.97 < m < 1$, providing little order-sense improvement over uncoded transmission.

The authors of [18] point out the fact that their result does not rule out the possibility that there may exist a zero-measure (asymptotically in n) subset of codes that has sub-quadratic wiring area. One could try to extend the bisection-width[12] approach of [18] to establish a negative result (i.e., prove that no such zero-measure set exists). To establish a positive result, one could try the open problem mentioned at the end of Section V-A2, namely, construct a graph-drawing algorithm that yields sub-quadratic crossing numbers for (even some classes of) semi-regular graphs. In any case, a proof is needed and heuristics such as those used in [64] (even if they work well in practice) cannot establish guarantees.

The simulation-based estimates of decoding power presented in Section VI confirm that coding can be useful for minimizing total power, even at short-distances. For instance, they predict that regular-LDPC codes with simple message-passing decoders can achieve lower bit-error probabilities than uncoded transmission in short distance settings, while still consuming the same total power (even at distances as low as 2 meters). However, in these regimes, it is possible that

[12]The crossing number $cr(\mathcal{G})$ and bisection width $bw(\mathcal{G})$ of a bounded-degree graph $\mathcal{G} = \{V, E\}$ are related by the inequality $cr(\mathcal{G}) + \Theta(|V|) = \Omega\left(bw^2(\mathcal{G})\right)$ [66, Theorem 2.1].

"classical" algebraic codes (e.g., Hamming or Reed-Solomon codes [67]) might be even more efficient, hence, they need to be examined as well.

Finally, the results of Section VI point to a new problem, that of "energy-adaptive codes". The suggestion from these results is that the code should be adapted to changing error-probabilities and distances. Can a single code, with a single piece of reconfigurable decoding hardware, enable adaptation of transmit and circuit power to minimize total energy? Indeed, some follow-up work [68] indicates this is possible, and it could be a promising direction for future work.

Acknowledgements

This work was supported in part by the NSF Center for Science of Information (CSoI) NSF CCF-0939370, as well as a seed grant on "Understanding Information-Energy Interactions" from the NSF CSoI. This work is also supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. Grover's research supported by NSF CCF-1350314 (NSF CAREER), and NSF ECCS-1343324 (NSF EARS) awards. We thank Anant Sahai, Subhasish Mitra, Yang Wen, Haewon Jeong, and Jianying Luo for stimulating discussions, and Jose Moura for code constructions that we started our simulations with. We thank the students, faculty, staff and sponsors of the Berkeley Wireless Research Center and Wireless Foundations group at Berkeley. In particular, Brian Richards assisted with the simulation flow and Tsung-Te Liu advised us on modeling circuits. Finally, we thank the anonymous reviewers whose comments helped us greatly in improving the manuscript.

Appendix A
Proof of Lemma 2

Proof of Lemma 2: First, note that the $\Omega(\cdot)$ expression in Lemma 2 contains two variables: $\frac{1}{P_e}$ and P_T. We analyze blocklength as a function $n : [2,\infty) \times \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 1}$ (Definition 3). First consider the case where $d_c > d_v \geq 3$. Because the codes considered are binary and linear, the channel is memoryless, binary-input, and output-symmetric, and the decoding computations are symmetric with respect to codewords, we can assume without loss of generality that the all-zero codeword was transmitted [25], [40, Page 22]. In [40, Page 37] it is shown that for *any* binary regular-LDPC code with $d_c > d_v \geq 3$ used to transmit over an AWGN channel, the probability that any iterative message-passing decoder incorrectly decides on pseudo-codeword[13] ω when the all-zero codeword was transmitted is

$$P_{0\to\omega} \geq \mathbb{Q}\left(\sqrt{2\frac{E_s}{N_0} w_p^{\text{AWGN}}(\omega)}\right), \tag{19}$$

where $w_p^{\text{AWGN}}(\omega)$ is called the AWGN pseudoweight of ω and is defined [40, Definition 31] as

$$w_p^{\text{AWGN}}(\omega) = \begin{cases} \dfrac{||\omega||_1^2}{||\omega||_2^2} = \dfrac{\left(\sum_{i\in[n]}\omega_i^2\right)^2}{\sum_{i\in[n]}\omega_i^2} & \text{if } \omega \neq 0. \\ 0 & \text{if } \omega = 0. \end{cases}$$

While the channel model of Section II-A assumed a hard-decision on the AWGN-channel outputs, (19) holds even when log-likelihood ratios (which have no loss in optimality) are used in message-passing [40]. Hence, we can use (19) to obtain a lower bound for *any* message-passing decoder. The *minimum* pseudoweight $w_p^{\text{AWGN,min}}$ of a parity-check matrix of a given code is defined as the minimum AWGN pseudoweight over all *nonzero* pseudo-codewords of the code [40, Definition 37]. For any regular-LDPC code of blocklength n with $d_v \geq 3$, the minimum AWGN pseudoweight is upper-bounded as [40, Proposition 49], [69, Theorem 7]:

$$w_p^{\text{AWGN,min}}(\omega) \leq \left(\frac{d_v(d_v-1)}{(d_v-2)}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}. \tag{20}$$

Therefore, lower bounding the word-error probability P_e^{word} by the pairwise error-probability and using (20) in (19):

$$P_e^{word} \geq P_{0\to\omega} \geq \mathbb{Q}\left(\sqrt{2\frac{E_s}{N_0}\left(\frac{d_v(d_v-1)}{(d_v-2)}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}}\right). \tag{21}$$

Using our notation from Section II-A, $\frac{E_s}{N_0} = \eta P_T$, and trivially bounding bit-error probability P_e [70, Eqn. (2)]:

$$\begin{aligned} P_e &\geq \frac{P_e^{word}}{n} \\ &\geq \frac{\mathbb{Q}\left(\sqrt{2\eta P_T\left(\frac{d_v(d_v-1)}{(d_v-2)}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}}\right)}{n} \\ &\overset{\bullet}{>} \frac{\frac{1}{n}e^{-\eta P_T\left(\frac{d_v(d_v-1)}{d_v-2}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}}}{\left(n^{\frac{1}{1+\frac{\log(d_c-1)}{\log(d_v-1)}}} 2\sqrt{\pi\eta P_T}\left(\frac{d_v(d_v-1)}{(d_v-2)}\right) + \sqrt{\frac{\pi}{\eta P_T}}\frac{d_v-2}{d_v(d_v-1)}\right)} \\ &\overset{(*)}{>} \frac{e^{-\eta P_T\left(\frac{d_v(d_v-1)}{d_v-2}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}}}{n^{1+\frac{\log(d_v-1)}{\log(d_v-1)(d_c-1)}}\left(\frac{d_v(d_v-1)}{(d_v-2)}\right)\left(2\sqrt{\pi\eta P_T}+\sqrt{\frac{\pi}{\eta P_T}}\right)} \end{aligned} \tag{22}$$

where $(\bullet)$ holds because of (1) and $n \geq 1$, and $(*)$ holds because $n \geq 1$ and $d_v(d_v-1) > (d_v-2)$. It follows that whenever $P_T \geq \frac{1}{\eta}$,

$$\begin{aligned} P_e &> \frac{e^{-\eta P_T\left(\frac{d_v(d_v-1)}{d_v-2}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}}}{n^{1+\frac{\log(d_v-1)}{\log(d_v-1)(d_c-1)}}\left(3\frac{d_v(d_v-1)}{(d_v-2)}\sqrt{\pi\eta P_T}\right)} \\ &\overset{(\star)}{>} \frac{e^{-\eta P_T(1+9\pi)\left(\frac{d_v(d_v-1)}{d_v-2}\right)^2 n^{\frac{2\log(d_v-1)}{\log(d_v-1)(d_c-1)}}}}{n} \end{aligned} \tag{23}$$

where $(\star)$ holds because $e^{-x^2} < \frac{1}{x}$ for all $x \geq 0$. Inverting

[13] Defined in [40] as an error-pattern for a given code $\mathcal{C}$, such that the lifting of the error-pattern is a codeword of the binary code corresponding to some finite graph-cover of the Tanner graph of $\mathcal{C}$. It is explained in [40] that no "locally operating" iterative message-passing decoding algorithm ("locally operating" subsumes all algorithms satisfying the assumptions of Section II-B) can distinguish between codewords and pseudo-codewords.

both sides of (23), taking $\log(\cdot)$ and then simplifying,

$$\begin{aligned}\frac{\log\frac{1}{P_e}}{\eta P_T(1+9\pi)\left(\frac{d_v(d_v-1)}{(d_v-2)}\right)^2} &< n^{\frac{2}{1+\frac{\log(d_c-1)}{\log(d_v-1)}}} \\ &+ \frac{\left(\frac{(d_v-2)}{d_v(d_v-1)}\right)^2\log n}{\eta P_T(1+9\pi)} \\ &\overset{P_T\geq\frac{1}{\eta}}{<} n^{\frac{2}{1+\frac{\log(d_c-1)}{\log(d_v-1)}}}+\log n \end{aligned} \tag{24}$$

We have shown that (24) holds for any P_e and any $P_T \geq \frac{1}{\eta}$, hence ignoring the non-dominating term on the RHS and then raising both sides to the power $\frac{\log(d_v-1)(d_c-1)}{2\log(d_v-1)}$, we get the desired result. For the case when $d_v = 2$, because the minimum distance of regular-LDPC codes with $d_v = 2$ is at most $2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}$ (see [29, Theorem 2.5]), the pairwise error-probability that a minimum-weight nonzero *codeword* x' is decoded when the all-zero codeword was transmitted is

$$P_{0\to x'} \geq \mathbb{Q}\left(\sqrt{2\frac{E_s}{N_0}\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}\right), \tag{25}$$

Replacing $\frac{E_s}{N_0}$ by ηP_T, the word-error probability is

$$P_e^{word} \geq P_{0\to x'} = \mathbb{Q}\left(\sqrt{2\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}\right). \tag{26}$$

Then applying an identical analysis to the $d_v > 3$ case, for any bit-error probability P_e:

$$\begin{aligned}P_e &\geq \frac{\frac{1}{n}e^{-\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}}{\left(2\sqrt{\pi\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}+\sqrt{\frac{\pi}{\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}}\right)} \\ &\overset{n\geq 1}{>} \frac{\frac{1}{n}e^{-\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}}{\left(2\sqrt{\pi\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}+\sqrt{\frac{\pi\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}{\eta P_T}}\right)}.\end{aligned}$$

Then for any $P_T \geq \frac{1}{\eta}$, we also have

$$P_e > \frac{e^{-\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}}{3n\sqrt{\pi\eta P_T\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}} \overset{(\star)}{>} \frac{e^{-\eta P_T(1+9\pi)\left(2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}\right)}}{n}, \tag{27}$$

where $(\star)$ holds because $e^{-x^2} < \frac{1}{x}$ for all $x \geq 0$. Inverting both sides of (27), taking $\log(\cdot)$ and then simplifying,

$$\begin{aligned}\frac{\log\frac{1}{P_e}}{\eta P_T(1+9\pi)} &< 2+\frac{2\log\frac{n}{2}}{\log(d_c-1)}+\frac{\log n}{\eta P_T(1+9\pi)} \\ \frac{\log\frac{1}{P_e}}{\eta P_T(1+9\pi)} &\overset{P_T\geq\frac{1}{\eta}}{<} 2+\frac{2\log n-2\log 2}{\log(d_c-1)}+\log n \\ &\overset{n\geq 1}{\leq} 2+\frac{2\log n-2\log 2}{\log(d_c-1)}+2\log n \\ &< \left(2+\frac{2}{\log(d_c-1)}\right)(1+\log n).\end{aligned} \tag{28}$$

Dividing both sides of (28) by $\left(2+\frac{2}{\log(d_c-1)}\right)$, taking $e^{(\cdot)}$ on both sides, and simplifying:

$$n > \frac{1}{e}\left(\frac{1}{P_e}\right)^{\frac{1}{\eta P_T(1+9\pi)\left(2+\frac{2}{\log(d_c-1)}\right)}}, \tag{29}$$

which completes the proof of Lemma 2. □

Appendix B
Proof of Lemma 3

Proof of Lemma 3: First, note the $\Theta(\cdot)$ expression in Lemma 3 contains two variables: $\frac{1}{P_e}$ and P_T. We analyze the minimum number of independent iterations as a function $N_{\text{iter}} : [2,\infty) \times [\Delta_A,\infty) \to \mathbb{R}^{\geq 0}$ (Definition 3), where $\Delta_A > 0$ is the transmit power for which p_0 is exactly the *threshold* for decoding over the BSC [29, Section 4.3], [25]. Explicitly, if σ_A is the threshold for Gallager-A decoding over the BSC, $\mathbb{Q}\left(\sqrt{2\eta\Delta_A}\right) = \sigma_A$.

Note when $P_T < \Delta_A$, it is not possible to force $P_e \to 0$, hence N_{iter} will be infinite for all P_e below some constant [25]. No further analysis is needed for such low transmit powers, since all P_e above said constant can be achieved with $\Theta(1)$ transmit power and $\mathcal{O}(1)$ decoding iterations. From [25, Eqn. (6)], the bit-error probability after the ith decoding iteration, p_i, is

$$\begin{aligned}p_i &= p_0 - p_0\left[\frac{1+(1-2p_{i-1})^{dc-1}}{2}\right]^{d_v-1} \\ &+ (1-p_0)\left[\frac{1-(1-2p_{i-1})^{dc-1}}{2}\right]^{d_v-1}.\end{aligned} \tag{30}$$

Since the RHS of (30) is differentiable with respect to (w.r.t.) p_{i-1}, by Taylor's Theorem there exists a real function $R_1(x)$ with $\lim_{x\to 0} R_1(x) = 0$ such that:

$$p_i = p_0(d_v-1)(d_c-1)p_{i-1} + R_1(p_{i-1}). \tag{31}$$

The RHS of (31) is the first-order MacLaurin expansion of p_i. Further, because the RHS of (30) is a polynomial in p_{i-1}, it is twice continuously differentiable and by the mean value theorem the remainder term $R_1(p_{i-1})$ has Lagrange form:

$$R_1(p_{i-1}) = \frac{1}{2}\frac{d^2p_i(x^*)}{dp_{i-1}^2}p_{i-1}^2, \tag{32}$$

where $x^* \in (0, p_{i-1})$. It can be verified that the second derivative of p_i w.r.t. p_{i-1} is minimized at $p_{i-1} = 0$ and maximized at $p_{i-1} = \frac{1}{2}$. Solving for both cases and plugging into (32), we find

$$\begin{aligned}-p_0(d_v-1)(d_c-1)\left[\frac{(d_v-2)(d_c-1)}{2}+(d_c-2)\right]p_{i-1}^2 \\ \leq R_1 \leq 0\end{aligned} \tag{33}$$

Plugging (33) into (31) and applying the RHS recursively, the bit-error probability after ith decoding iteration p_i is:

$$\begin{aligned}p_0(d_v-1)(d_c-1)p_{i-1}\Bigg[1-p_{i-1}\Bigg(\frac{(d_v-2)(d_c-1)}{2} \\ +(d_c-2)\Bigg)\Bigg] \leq p_i \leq [p_0(d_v-1)(d_c-1)]^i.\end{aligned} \tag{34}$$

Now, choose an arbitrary $0 < \delta < \frac{1}{2}$ and choose P_T (thereby p_0 as well). As explained in [29, Section 4.3], since we are operating above the threshold, we are guaranteed that $p_i < p_{i-1} \leq p_0$ and $p_i \overset{i\to\infty}{\to} 0$. Thus, for sufficiently small p_i (thereby small p_{i-1}) *or* sufficiently large P_T (thereby small p_0), (34) becomes:

$$p_0(d_v-1)(d_c-1)(1-\delta)p_{i-1} \leq p_i \leq \left[p_0(d_v-1)(d_c-1)\right]^i \tag{35}$$

Applying the relation on the LHS of (35) recursively

$$\begin{aligned}
&p_0\left[\frac{p_0}{2}(d_v-1)(d_c-1)\right]^i \overset{\delta<\frac{1}{2}}{\leq} p_i \leq \left[p_0(d_v-1)(d_c-1)\right]^i \\
&\frac{\sqrt{\frac{\eta}{\pi}P_T}e^{-\eta P_T}}{(2\eta P_T+1)}\left[\frac{(d_v-1)(d_c-1)\sqrt{\frac{\eta}{\pi}P_T}e^{-\eta P_T}}{2\left(2\eta P_T+1\right)}\right]^i \overset{(1)}{\leq} p_i \\
&\overset{(1)}{\leq}\left[(d_v-1)(d_c-1)\frac{e^{-\eta P_T}}{\sqrt{4\pi\eta P_T}}\right]^i.
\end{aligned} \tag{36}$$

Inverting all sides of (36), taking $\log(\cdot)$ on all sides, replacing p_i by P_e and i by N_{iter}, and dividing all sides by ηP_T:

$$\begin{aligned}
&N_{\text{iter}}\left[1+\frac{\log\eta P_T}{2\eta P_T}-\frac{\log(d_v-1)(d_c-1)}{\eta P_T}+\frac{\log 2\sqrt{\pi}}{\eta P_T}\right] \\
&\leq \frac{\log\frac{1}{P_e}}{\eta P_T} \leq N_{\text{iter}}\left[1-\frac{\log\frac{\eta}{\pi}P_T}{2\eta P_T}-\frac{\log 0.5(d_v-1)(d_c-1)}{\eta P_T}\right. \\
&\left.+\frac{\log\left(2\eta P_T+1\right)}{\eta P_T}\right]+1+\frac{\log\left(2\eta P_T+1\right)}{\eta P_T}-\frac{\log\frac{\eta}{\pi}P_T}{2\eta P_T}.
\end{aligned} \tag{37}$$

We have shown that (37) holds for any choice of P_T as long as P_e is sufficiently small, which completes the proof of the constant P_T result. Next, set $P_T \geq \max\{\frac{2\log(d_v-1)(d_c-1)}{\eta}, \frac{\pi}{\eta}\}$. As explained above, (37) also holds for any P_e as long as P_T is sufficiently large. In this case (37) simplifies to

$$\begin{aligned}
&N_{\text{iter}}\left[1-\frac{\log(d_v-1)(d_c-1)}{\eta P_T}\right] \overset{P_T>\frac{1}{\eta}>0}{\leq} \frac{\log\frac{1}{P_e}}{\eta P_T} \\
&\overset{P_T\geq\frac{\pi}{\eta}>\frac{1}{\eta};d_c>d_v\geq 2}{\leq} N_{\text{iter}}\left[1+\log 3\right]+1+\log 3 \\
&\frac{1}{2}N_{\text{iter}} \overset{P_T\geq\frac{2\log(d_v-1)(d_c-1)}{\eta}}{\leq} \frac{\log\frac{1}{P_e}}{\eta P_T} < \left[1+\log 3\right]N_{\text{iter}}+3,
\end{aligned} \tag{38}$$

which completes the proof of the Lemma. □

Appendix C
Proof of Lemma 4

Proof of Lemma 4: We analyze the number of independent iterations as a function $N_{\text{iter}} : [2,\infty) \to \mathbb{R}^{\geq 0}$, since even in the second case, the transmit power is a function of $\frac{1}{P_e}$. Let $\Delta_B > 0$ be the transmit power for which p_0 is exactly the *threshold* for decoding over the BSC [29, Section 4.3], [25]. As explained in the proof of Lemma 3, we need not consider cases where $P_T < \Delta_B$. Using [29, Eqn. 4.15], for d_v odd, the bit-error probability after the ith decoding iteration follows

$$\begin{aligned}
p_i = p_0 &- \frac{p_0}{2^{dv-1}}\sum_{m=\frac{d_v-1}{2}}^{d_v-1}\binom{d_v-1}{m}\left[1+(1-2p_{i-1})^{d_c-1}\right]^m \\
&\times\left[1-(1-2p_{i-1})^{d_c-1}\right]^{d_v-1-m} \\
&+\frac{1-p_0}{2^{dv-1}}\sum_{m=\frac{d_v-1}{2}}^{d_v-1}\binom{d_v-1}{m}\left[1-(1-2p_{i-1})^{d_c-1}\right]^m \\
&\times\left[1+(1-2p_{i-1})^{d_c-1}\right]^{d_v-1-m}.
\end{aligned} \tag{39}$$

The RHS of (39) is a polynomial in p_{i-1} and the $\frac{d_v-1}{2}$th order Maclaurin expansion is

$$p_i = p_0\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)^{\frac{d_v-1}{2}}p_{i-1}^{\frac{d_v-1}{2}}+R_B(p_{i-1}). \tag{40}$$

Because the RHS of (39) is a polynomial, by the mean value theorem, the remainder has a Lagrange form (where $x^* \in (0, p_{i-1})$):

$$R_B(p_{i-1}) = \frac{1}{(\frac{d_v+1}{2})!}\frac{d^{\frac{d_v+1}{2}}p_i(x^*)}{dp_{i-1}^{\frac{d_v+1}{2}}}p_{i-1}^{\frac{d_v+1}{2}}. \tag{41}$$

The $\frac{d_v+1}{2}$th derivative of p_i is another polynomial; therefore it must be bounded on the bounded interval $[0, \frac{1}{2}]$ and

$$-c_l^B p_{i-1}^{\frac{d_v+1}{2}} \leq R_B(p_{i-1}) \leq c_u^B p_{i-1}^{\frac{d_v+1}{2}}, \tag{42}$$

for some constants $c_l^B, c_u^B > 0$. Then choose P_T. Since we exceed the decoding threshold [25], $p_i < p_{i-1} \leq p_0$. Now, take i to be the final iteration. Since we assumed $\lim_{P_e\to 0}\frac{P_T}{\log\frac{1}{P_e}} = 0$, we will also have $\lim_{P_e\to 0}\frac{p_{i-1}}{p_0} = 0$ (the number of decoding iterations used in the coding strategy eventually exceeds 1 as $P_e \to 0$). Hence, for sufficiently small p_i (thereby small p_{i-1}),

$$\begin{aligned}
&-\frac{1}{2}p_0\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)^{\frac{d_v-1}{2}}p_{i-1}^{\frac{d_v-1}{2}} \leq R_B(p_{i-1}) \\
&\leq \frac{1}{2}p_0\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)^{\frac{d_v-1}{2}}p_{i-1}^{\frac{d_v-1}{2}}.
\end{aligned} \tag{43}$$

Plugging (43) into (40), we have

$$\begin{aligned}
&p_0\left[\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)^{\frac{d_v-1}{2}}\frac{1}{2}\right]p_{i-1}^{\frac{d_v-1}{2}} \leq p_i \\
&\leq p_0\left[\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)^{\frac{d_v-1}{2}}\frac{3}{2}\right]p_{i-1}^{\frac{d_v-1}{2}}.
\end{aligned} \tag{44}$$

Applying (44) recursively, we obtain

$$\begin{aligned}
&\left[p_0^{1+\cdots+\left(\frac{d_v-1}{2}\right)^i}\left(\binom{d_v-1}{\frac{d_v-1}{2}}\frac{1}{2}\right)^{1+\cdots+\left(\frac{d_v-1}{2}\right)^{i-1}}\right. \\
&\left.\times(d_c-1)^{\left(\frac{d_v-1}{2}\right)+\cdots+\left(\frac{d_v-1}{2}\right)^i}\right] \leq p_i \\
&\leq\left[p_0^{1+\cdots+\left(\frac{d_v-1}{2}\right)^i}\left(\binom{d_v-1}{\frac{d_v-1}{2}}\frac{3}{2}\right)^{1+\cdots+\left(\frac{d_v-1}{2}\right)^{i-1}}\right. \\
&\left.\times(d_c-1)^{\left(\frac{d_v-1}{2}\right)+\cdots+\left(\frac{d_v-1}{2}\right)^i}\right].
\end{aligned} \tag{45}$$

Loosening the LHS and RHS of (45) and grouping like-terms

we have

$$\left[p_0^{1+\cdots+\left(\frac{d_v-1}{2}\right)^i}\left(\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)\frac{1}{2}\right)^{1+\cdots+\left(\frac{d_v-1}{2}\right)^{i-1}}\right]$$
$$\overset{(i\geq 0;d_c>2)}{\leq} p_i \overset{(p_0\leq 1;d_c>2)}{\leq} \left(p_0\binom{d_v-1}{\frac{d_v-1}{2}}\frac{3}{2}\right)^{1+\cdots+\left(\frac{d_v-1}{2}\right)^{i-1}}$$
$$\times (d_c-1)^{1+\cdots+\left(\frac{d_v-1}{2}\right)^i}. \quad (46)$$

Simplifying geometric progressions, inverting all sides of (46), taking $\log(\cdot)$ on all sides, and replacing p_i by P_e and i by N_{iter}:

$$\frac{\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}}-1}{\left(\frac{d_v-1}{2}\right)-1}\left[\log\frac{1}{p_0}+\log\frac{2}{3\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)}\right]\leq\log\frac{1}{P_e}$$
$$\leq\frac{\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}+1}-1}{\left(\frac{d_v-1}{2}\right)-1}\log\frac{1}{p_0}$$
$$+\frac{\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}}-1}{\left(\frac{d_v-1}{2}\right)-1}\log\frac{2}{\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)}. \quad (47)$$

Applying (1) on p_0 terms in (47), dividing all sides by ηP_T, loosening the RHS by ignoring negative terms, and simplifying:

$$\frac{\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}}-1}{(d_v-3)}\left[2+\frac{\log 4\pi\eta P_T-2\log\frac{3}{2}\binom{d_v-1}{\frac{d_v-1}{2}}(d_c-1)}{\eta P_T}\right]$$
$$\leq\frac{\log\frac{1}{P_e}}{\eta P_T}\leq\frac{\log(d_c-1)}{\eta P_T}$$
$$+\frac{\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}+1}}{(d_v-3)}\left[2+\frac{2\log\left(\sqrt{4\pi\eta P_T}+\sqrt{\frac{\pi}{\eta P_T}}\right)}{\eta P_T}\right]. \quad (48)$$

We have shown that (48) holds for any P_T as long as P_e is sufficiently small. Thus, treating P_T as a constant in (48) and taking $\log(\cdot)$ on all sides completes the proof of the fixed P_T result. For the other case, consider the limits of the leftmost and rightmost side of (48) as $P_T\to\infty$. For any $\epsilon>0$, for sufficiently large P_T the following holds:

$$\left(\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}}-1\right)\frac{2-\epsilon}{(d_v-3)}\leq\frac{\log\frac{1}{P_e}}{\eta P_T}$$
$$\leq\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}+1}\frac{2+\epsilon}{(d_v-3)}. \quad (49)$$

Taking $\log(\cdot)$ on all sides of (49) and simplifying, we obtain:

$$\log\left(\left(\frac{d_v-1}{2}\right)^{N_{\text{iter}}}-1\right)+\log\frac{2-\epsilon}{(d_v-3)}\leq\log\frac{\log\frac{1}{P_e}}{\eta P_T}$$
$$\leq(N_{\text{iter}}+1)\log\left(\frac{d_v-1}{2}\right)+\log\frac{2+\epsilon}{(d_v-3)}, \quad (50)$$

which is equivalent to the desired result. □

Appendix D
Proof of Theorem 3

Proof of Theorem 3: Since we are proving a lower bound, we can restrict ourselves to the case where $d_v\geq 3$ without loss of generality (the decoding power when $d_v=2$ grows exponentially faster). Via Lemma 2 and Lemma 5, the total power required for a (d_v,d_c)-regular LDPC code and any iterative message-passing decoder to achieve bit-error probability P_e under the Wire Model is

$$P_{\text{total}}=\Omega\left(P_T+\left(\frac{\log\frac{1}{P_e}}{\eta P_T(1+9\pi)\left(\frac{d_v(d_v-1)}{d_v-2}\right)^2}\right)^{\frac{1+\frac{\log(d_c-1)}{\log(d_v-1)}}{2}}\right) \quad (51)$$

First, it follows from (22) that if P_T is kept fixed while $P_e\to 0$, the total power (and the decoding power) diverges as

$$P_{\text{total,bdd P}_\text{T}}=\Omega\left(\log^{\frac{1+\frac{\log(d_c-1)}{\log(d_v-1)}}{2}}\frac{1}{P_e}\right). \quad (52)$$

The exponent of $\log\frac{1}{P_e}$ in (52) is always greater than 1 since $d_c>d_v$ for any regular-LDPC code. Next, differentiating the expressions inside the $\Omega(\cdot)$ of (51) w.r.t. P_T, setting to zero, and substituting the minimizing transmit power into (51) we find that the minimum total power is:

$$P_{\text{total,min}}=\Omega\left(\log^{\frac{\frac{1}{2}}{1+\frac{\log(d_c-1)}{\log(d_v-1)}}}\frac{1}{P_e}\right), \quad (53)$$

which completes the proof of the theorem. □

Appendix E
Proof of Theorem 4

Proof of Theorem 4: Let $N_{\text{iter}}^{(P_e)}$ denote the minimum number of independent Gallager-A decoding iterations required to achieve bit-error probability P_e. Via Theorem 2, the total power is lower bounded by $P_{\text{total}}=P_T+\Omega\left(e^{\gamma N_{\text{iter}}^{(P_e)}}\right)$. It follows from Lemma 3 that if P_T is kept fixed as $P_e\to 0$, then the required decoding power diverges atleast as fast as a power of $\frac{1}{P_e}$, which is exponentially larger than the power required for uncoded transmission. If instead the transmit power is allowed to vary, it follows from Lemma 3 that

$$P_{\text{total}} = \Omega\left(P_T+\left(\frac{1}{P_e}\right)^{\frac{\gamma}{\eta P_T}}\right). \quad (54)$$
$$P_{\text{total}} = \mathcal{O}\left(P_T+\left(\frac{1}{P_e}\right)^{\frac{2\gamma}{\eta P_T}}\right) \quad (55)$$

In order to find the optimizing transmit power, let $L_{P_e}(P_T)$ denote the function in the $\Omega(\cdot)$ expression of (54) and let $U_{P_e}(P_T)$ denote the function in the $\mathcal{O}(\cdot)$ expression of (55):

$$L_{P_e}(P_T) = P_T+\left(\frac{1}{P_e}\right)^{\frac{\gamma}{\eta P_T}} \quad (56)$$
$$U_{P_e}(P_T) = P_T+\left(\frac{1}{P_e}\right)^{\frac{2\gamma}{\eta P_T}}. \quad (57)$$

We start by analyzing the lower bound. To find the P_T which

minimizes L_{P_e}, we differentiate L_{P_e} and set it to 0

$$\begin{aligned}\frac{dL_{P_e}}{dP_T} &= 1 - e^{\frac{\gamma \log \frac{1}{P_e}}{\eta P_T}} \frac{\gamma \log \frac{1}{P_e}}{\eta P_T^2} = 0 \\ \Rightarrow \frac{P_T^2}{\frac{\gamma \log \frac{1}{P_e}}{\eta}} &= e^{\frac{\gamma \log \frac{1}{P_e}}{\eta P_T}}. \end{aligned} \tag{58}$$

Now, let $\mathcal{P} = \frac{P_T}{\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}}$. Substituting into (58), we get

$$\mathcal{P}^2 = e^{\frac{\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}}{\mathcal{P}}} \Rightarrow 2\mathcal{P} \log \mathcal{P} = \sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}} \tag{59}$$

$$\Rightarrow \log \mathcal{P} e^{\log \mathcal{P}} = \frac{1}{2}\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}. \tag{60}$$

The positive, real valued solution to (60) is given by the principal branch $W_0(\cdot)$ of the Lambert W function [71]. Explicitly, when $x, z \in \mathbb{R}^{\geq 0}$ satisfy the relation $x = ze^z$, we say $z = W_0(x)$. Hence we can write

$$\log \mathcal{P} = W_0\left(\frac{1}{2}\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}\right) \tag{61}$$

$$\Rightarrow \mathcal{P} = \frac{\log \mathcal{P} e^{\log \mathcal{P}}}{\log \mathcal{P}} \overset{(60)(61)}{=} \frac{\frac{1}{2}\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}}{W_0\left(\frac{1}{2}\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}\right)}. \tag{62}$$

Rewriting $\mathcal{P}$ in terms of P_T we find the optimizing transmit power

$$P_T^* = \frac{\frac{\gamma \log \frac{1}{P_e}}{\eta}}{2W_0\left(\frac{1}{2}\sqrt{\frac{\gamma \log \frac{1}{P_e}}{\eta}}\right)}. \tag{63}$$

The first two terms in the asymptotic expansion of $W_0(x)$ as $x \to \infty$ are $\log(x) - \log\log(x)$ [71]. In fact, $\forall x \geq e$ [72]:

$$\log(x) - \log\log(x) \leq W_0(x) \leq \log(x) - \frac{1}{2}\log\log(x). \tag{64}$$

Using (64) in (63), and ignoring constant terms in the resulting denominator, the optimizing transmit power is

$$P_T^* = \Theta\left(\frac{\frac{\gamma}{\eta}\log \frac{1}{P_e}}{\log\log \frac{1}{P_e} - 2\log\log\log^{\frac{1}{2}} \frac{1}{P_e}}\right).$$

Plugging back into L_{P_e} in (56), and ignoring non-dominating terms, we get the lower bound. An identical analysis of U_{P_e} in (57) gives the upper bound and completes the proof. □

Appendix F
Proof of Theorem 5

Proof of Theorem 5: Via Theorem 2 and Lemma 4, if the transmit power is kept fixed as $P_e \to 0$, the total power is

$$P_{\text{total,bdd P}_\text{T}} = \Omega\left(P_T + \log^{\gamma} \frac{1}{P_e}\right). \tag{65}$$

If instead P_T is allowed to scale as a function of P_e,

$$P_{\text{total}} \overset{Theorem\ 2}{=} \Omega\left(P_T + e^{\gamma N_{\text{iter}}^{(P_e)}}\right) \tag{66}$$

$$\overset{Lemma\ 4}{=} \Omega\left(P_T + e^{\frac{\gamma}{\log\left(\frac{d_v-1}{2}\right)} \log \frac{\log \frac{1}{P_e}}{P_T}}\right) \tag{67}$$

$$= \Omega\left(P_T + \left(\frac{\log \frac{1}{P_e}}{P_T}\right)^{\frac{\gamma}{\log\left(\frac{d_v-1}{2}\right)}}\right) \tag{68}$$

Using the upper bound from Theorem 2,

$$P_{\text{total}} = \mathcal{O}\left(P_T + \left(\frac{\log \frac{1}{P_e}}{P_T}\right)^{\frac{2\gamma}{\log\left(\frac{d_v-1}{2}\right)}}\right). \tag{69}$$

Then, considering the bounds on γ in Theorem 2, we examine the exponents of $\log \frac{1}{P_e}$ in (65) and $\frac{\log \frac{1}{P_e}}{P_T}$ in (68):

$$\begin{aligned}\gamma &\geq \log(d_v - 1) + \log(d_c - 1) \overset{d_c > d_v > 3}{\geq} \log 12 \\ \Rightarrow \frac{\gamma}{\log\left(\frac{d_v-1}{2}\right)} &> \frac{1 + \frac{\log(d_c-1)}{\log(d_v-1)}}{1 - \frac{\log 2}{\log(d_v-1)}} \overset{d_c > d_v > 3}{\geq} 2. \end{aligned} \tag{70}$$

It follows that if the transmit power is kept fixed even as $P_e \to 0$, the total power diverges as $P_{\text{total,bdd P}_\text{T}} = \Omega\left(\log^{2.48} \frac{1}{P_e}\right)$. Moving to the unbounded case, substituting (70) back into (68), we obtain:

$$P_{\text{total}} = \Omega\left(P_T + \left(\frac{\log \frac{1}{P_e}}{P_T}\right)^2\right). \tag{71}$$

Differentiating the expression inside the $\Omega(\cdot)$ on the RHS of (71) w.r.t. P_T and setting to zero, the total power scales like:

$$P_{\text{total,min}} = \Omega\left(\log^{\frac{2}{3}} \frac{1}{P_e}\right).$$

This lower bound tightens when $d_v d_c \geq 4(d_v + d_c)$. Using Theorem 2 for this case,

$$P_{\text{total}} = \Omega\left(P_T + \left(\frac{\log \frac{1}{P_e}}{P_T}\right)^{\log 32}\right)$$

$$P_{\text{total,min}} \overset{\frac{\log 32}{\log 32+1} > \frac{31}{40}}{=} \Omega\left(\log^{\frac{31}{40}} \frac{1}{P_e}\right).$$

Moving to the upper bound, via Theorem 2, we find that the exponent of $\frac{\log \frac{1}{P_e}}{P_T}$ in (69) is

$$2\gamma \leq \frac{6\log(2d_v d_c + 1)}{\log(d_v - 1) - \log 2}. \tag{72}$$

Then substituting (72) into (69), we get the bound

$$P_{\text{total}} = \mathcal{O}\left(P_T + \left(\frac{\log \frac{1}{P_e}}{P_T}\right)^{\frac{6\log(2d_v d_c+1)}{\log(d_v-1)-\log 2}}\right). \tag{73}$$

Differentiating the expressions inside the $\mathcal{O}(\cdot)$ of (73) w.r.t. P_T and setting to zero, we obtain the upper bound. □

APPENDIX G
CAD FLOW DETAILS

The decoding implementation models are constructed in a hierarchical manner. First, behavioral verilog descriptions of variable and check nodes are mapped to standard cells using logic synthesis[14] and are then placed-and-routed using a physical design tool. The physical area of the individual circuits is obtained. Post-layout simulation is then performed, using extracted RC parasitics and typical corners for the Synopsys 32/28nm HVT CMOS process at a supply voltage of 0.78V. The critical-path delays of the variable-nodes $T_{\mathrm{VN}}(a, d_v)$, and check-nodes $T_{\mathrm{CN}}(a, d_c)$ for each decoding algorithm are obtained using post-layout static timing analysis with the parasitics included. Post-layout power analysis is performed to obtain the average power consumption of variable-nodes $P_{\mathrm{VN}}(a, d_v)$ and check-nodes $P_{\mathrm{CN}}(a, d_c)$ using a "virtual clock" of period $T_{\mathrm{VN}}(a, d_v)$ or $T_{\mathrm{CN}}(a, d_c)$, respectively, over a large number of uniformly random input patterns. In practice however, the amount of switching activity at the decoder depends on the number of errors in the received sequence over the channel, and it thereby depends on the parameters of the channel and communication system. For example, when the transmit power is large and/or the path-loss and noise are small, the expected number of errors in the received sequence is small and the switching activity caused by bit-flips may be much smaller than these simulations indicate. Nevertheless, we assume (with slight overestimation) that the averaged power numbers hold for the various check-nodes and variable-nodes.

APPENDIX H
CIRCUIT MODEL FOR CRITICAL-PATH DELAY

It is assumed that all decoders operate at the minimum clock period $T_{\mathrm{CLK}}(a, g, d_v, d_c)$ for which timing would be met at the 0.78V supply voltage. This minimum allowable clock period that meets timing in flip-flop based synchronous circuits is bounded by the setup time constraint [19] for each flip-flop. The setup time is the minimum time it takes the incoming data to a flip-flops to propagate through the input stages of the flip-flop. The critical path for a full decoding iteration consists of a CLK-Q delay of a message passing flip-flop inside a variable-node, then an interconnect delay, then a check-node delay $T_{\mathrm{CN}}(a, d_c)$, then another interconnect delay, and finally a variable-node delay $T_{\mathrm{VN}}(a, d_v)$. In these models, the setup and the CLK-Q delay are accounted for in $T_{\mathrm{VN}}(a, d_v)$.

Interconnect delay is assumed to be linearly proportional to the resistance and capacitance of the interconnect, which depend on the length and width of interconnects. Estimating the length of interconnects requires an estimate of the decoder's physical dimensions. The total area of the decoder, A_{Decoder}, is estimated as a sum of check-node and variable-node areas, where the nodes are assumed to be placed in a square arrangement. Best-case and worst-case estimates for the average interconnect length $l_{\mathrm{wire}}(a, g, d_v, d_c)$ are obtained by the following equations [33]

$$l_{\mathrm{wire}}(a, g, d_v, d_c) = \begin{cases} A_{\mathrm{Decoder}}^{0.25} & \text{in best case.} \quad (74) \\ \dfrac{\sqrt{A_{\mathrm{Decoder}}}}{3} & \text{in worst case.} \quad (75) \end{cases}$$

Rigorous empirical and theoretical justification for the above estimates is provided in [33] where it is shown that (74) is a good approximation for highly-parallel logic and (75) is the average value for randomly-placed logic on a square array. Since the logic functions computed by the check-nodes and variable-nodes for the decoding algorithms considered in this paper are intrinsically parallel and we also assume the decoders are implemented in a fully-parallel manner, we used (74) for the results shown in this paper. However, we note that (75) could be a better approximation, depending on the code construction used.

Routing for decoders is assumed to use minimum-width wires on the lower 7 metal layers of the 9-layer CMOS process[15]. The average minimum width (w_{avg}), sheet resistance (R_{sq}), and capacitance per-unit-length (C_{unit}) [16] for these metal layers are calculated using design rule information [54] and are assumed as constants. Interconnect delay is then estimated assuming a distributed Elmore model [19]:

$$R_{\mathrm{wire}}(a, g, d_v, d_c) = R_{\mathrm{sq}} \times \frac{l_{\mathrm{wire}}(a, g, d_v, d_c)}{w_{\mathrm{avg}}} \quad (76)$$

$$C_{\mathrm{wire}}(a, g, d_v, d_c) = C_{\mathrm{unit}} \times l_{\mathrm{wire}}(a, g, d_v, d_c) \quad (77)$$

$$T_{\mathrm{wire}}(a, g, d_v, d_c) = \frac{R_{\mathrm{sq}} C_{\mathrm{unit}} l_{\mathrm{wire}}^2(a, g, d_v, d_c)}{2 w_{\mathrm{avg}}}. \quad (78)$$

APPENDIX I
CIRCUIT MODEL FOR COMPUTATION POWER

The power consumption of a logic gate consists of both dynamic power (which is proportional to the activity-factor at the input of the gate and the clock-frequency), and static power (which has no dependence on the activity-factor or the clock frequency) [19]. In post-layout simulation, the static power consumption of variable-nodes and check-nodes at 0.78V supply in a high threshold-voltage process is observed to be less than 1% of the total power in check-nodes and variable-nodes. Therefore, with little loss in accuracy, we treat the total power consumption of check-nodes and variable-nodes as dynamic power when considering the effect of clock-frequency scaling. Therefore, the power consumed after clock-frequency scaling in variable-nodes is $P_{\mathrm{VN}}(a, d_v) \times \frac{T_{\mathrm{VN}}(a, d_v)}{T_{\mathrm{CLK}}(a, g, d_v, d_c)}$ and in check-nodes it is $P_{\mathrm{CN}}(a, d_c) \times \frac{T_{\mathrm{VN}}(a, d_v)}{T_{\mathrm{CLK}}(a, g, d_v, d_c)}$.

APPENDIX J
CIRCUIT MODEL FOR INTERCONNECT POWER

Using the interconnect capacitance estimate $C_{\mathrm{wire}}(a, g, d_v, d_c)$ and clock period $T_{\mathrm{CLK}}(a, g, d_v, d_c)$ from Appendix H, and assuming an activity factor of $\frac{1}{2}$, the power consumed by a single message-passing

[14] The delay, power, area, and structure of synthesized logic depend on the constraints and mapping effort given as inputs to the synthesis tool. To allow for a fair comparison between codes of different degrees, we only specify constraints for minimum delay and minimum power and use the highest possible mapping effort for each node.

[15] The top two metal layers are often used to construct a global power grid for an entire chip.

[16] Including parallel-plate and fringing components [19].

interconnect ($P_{\text{wire}}(a, g, d_v, d_c)$) in the decoder is modeled using the formula for the dynamic power consumed in interconnects [19]:

$$P_{\text{wire}}(a, g, d_v, d_c) = \frac{C_{\text{wire}}(a, g, d_v, d_c) \times (0.78V)^2}{2T_{\text{CLK}}(a, g, d_v, d_c)}. \quad (79)$$

REFERENCES

[1] K. Ganesan, P. Grover, and A. Goldsmith, "How far are LDPC codes from fundamental limits on total power consumption?" in *50th Allerton Conference*, Monticello, IL, Oct. 2012, pp. 671–678.

[2] K. Ganesan, Y. Wen, P. Grover, A. Goldsmith, and J. Rabaey, "Choosing "green" codes by simulation-based modeling of implementations," in *Proc. of GLOBECOM*, Dec. 2012, pp. 3286–3292.

[3] V. Stojanović, "Channel-limited high-speed links: modeling, analysis and design," Ph.D. dissertation, Stanford University, 2004.

[4] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10GBASE-T ethernet LDPC decoder design with low error floors," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 843 –855, April 2010.

[5] *IEEE Std. 802.11ad-2012: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Very High Throughput in the 60 GHz Band," amendment*. IEEE, Dec. 2012.

[6] P. Grover, K. A. Woyach, and A. Sahai, "Towards a communication-theoretic understanding of system-level power consumption," *IEEE J. Select. Areas Comm.*, vol. 29, no. 8, pp. 1744–1755, Sept. 2011.

[7] C. Marcu et al., "A 90 nm CMOS low-power 60 GHz transceiver with integrated baseband circuitry," *IEEE J. Solid-State Circuits*, vol. 44, no. 12, pp. 3434–3447, Dec. 2009.

[8] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, Aug. 2008.

[9] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2007.

[10] S. Cui, A. Goldsmith, and A. Bahai, "Energy Constrained Modulation Optimization," *IEEE Trans. Wireless Comm.*, vol. 4, no. 5, pp. 1–11, Sept. 2005.

[11] P. Youssef-Massaad, M. Medard, and L. Zheng, "Impact of Processing Energy on the Capacity of Wireless Channels," in *Proc. of ISITA*, Parma, Italy, Oct. 2004.

[12] C. D. Thompson, "A complexity theory for VLSI," Ph.D. dissertation, Carnegie Mellon University, 1980.

[13] P. Grover, A. Goldsmith, and A. Sahai, "Fundamental limits on the power consumption of encoding and decoding," in *Proc. of ISIT*, Cambridge, MA, July 2012, pp. 2716–2720.

[14] P. Grover and A. Sahai, "Fundamental bounds on the interconnect complexity of decoder implementations," in *Proc. of 45th CISS*, Baltimore, MD, Mar. 2011, pp. 1–6.

[15] C. Blake and F. R. Kschischang, "Energy consumption of VLSI decoders," Dec. 2014, http://arxiv.org/abs/1412.4130.

[16] *IEEE Std. 802.3an-2006: "Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-T," amendment to IEEE Std. 802.3-2005*. IEEE, Sept. 2006.

[17] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The development of Turbo and LDPC codes for deep-space applications," *Proc. IEEE*, vol. 95, no. 11, pp. 2142–2156, Nov. 2007.

[18] C. Blake and F. R. Kschischang, "On the energy complexity of LDPC decoder circuits," Feb. 2015, http://arxiv.org/abs/1502.07999.

[19] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Prentice Hall, 2002.

[20] T. H. Lee, *The design of CMOS radio-frequency integrated circuits*. Cambridge university press, 2004.

[21] T. Sundström, B. Murmann, and C. Svensson, "Power dissipation bounds for high-speed Nyquist analog-to-digital converters," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 3, pp. 509–518, Mar. 2009.

[22] Y. Li, B. Bakkaloglu, and C. Chakrabarti, "A system level energy model and energy-quality evaluation for integrated transceiver front-ends," *IEEE Trans. VLSI*, vol. 15, no. 1, pp. 90–103, Jan. 2007.

[23] D. Knuth, *Art of Computer Programming Volume 1: Fundamental Algorithms*. Addison-Wesley Professional, 1997.

[24] G. Brassard and P. Bratley, *Fundamentals of algorithmics*. Prentice Hall, 1996.

[25] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[26] R. D. Gordon, "Values of mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument," *Ann. Math. Stat.*, vol. 12, no. 3, pp. 364–366, Sept. 1941.

[27] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498—519, Feb. 2001.

[28] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. Comm.*, vol. 53, no. 4, pp. 549–554, April 2005.

[29] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, MIT, 1960.

[30] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, "Predicting error floors of structured LDPC codes: Deterministic bounds and estimates," *IEEE J. Select. Areas Comm.*, vol. 27, no. 6, pp. 908–917, Aug. 2009.

[31] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[32] R. P. Brent and H.-T. Kung, "The chip complexity of binary arithmetic," in *Proc. of 12th STOC*, 1980, pp. 190–200.

[33] W. E. Donath, "Placement and average interconnection lengths of computer logic," *IEEE Trans. Circuits Syst. I*, vol. 26, no. 4, pp. 272–277, April 1979.

[34] P. Christie and D. Stroobandt, "The interpretation and application of Rent's rule," *IEEE Trans. VLSI*, vol. 8, no. 6, pp. 639–648, Dec. 2000.

[35] Wikipedia, "Back end of line," https://en.wikipedia.org/wiki/Back_end_of_line, 2015.

[36] C. D. Thompson, "VLSI design with multiple active layers," *Information Processing Letters*, vol. 21, no. 3, pp. 109–111, 1985.

[37] M. Aly et al., "Energy-efficient abundant-data computing: The N3XT 1,000X," *IEEE Computer*, Dec. 2015.

[38] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proc. of 41st DAC*, June 2004, pp. 868–873.

[39] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, 2003.

[40] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," Dec. 2005, http://arxiv.org/abs/cs/0512078.

[41] M. Lentmaier, D. V. Truhachev, K. S. Zigangirov, and D. J. Costello, "An analysis of the block error probability performance of iterative decoding," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.

[42] A. Shokrollahi, "Capacity achieving sequences," *Codes, Systems, Graphical Models*, vol. 123, pp. 153–166, 2001.

[43] C. E. Leiserson, "Area-efficient graph layouts (for VLSI)," in *Proc. of 21st Symp. on FOCS*, Oct. 1980, pp. 270–281.

[44] R. J. Lipton and R. Sedgewick, "Lower bounds for VLSI," in *Proc. of 13th Symp. on FOCS*, May 1981, pp. 300–307.

[45] F. T. Leighton, "Layouts for the shuffle-exchange graph and lower bound techniques for VLSI," Ph.D. dissertation, MIT, 1982.

[46] K. Zarankiewicz, "On a problem of P. Turán concerning graphs," *Fund. Math.*, vol. 41, pp. 137–145, 1954.

[47] J. Pach and G. Tóth, "Thirteen problems on crossing numbers," *Geombinatorics*, vol. 9, no. 4, pp. 194–207, 2000.

[48] J. Pach, J. Spencer, and G. Tóth, "New bounds on crossing numbers," *Dis. Comp. Geo.*, vol. 24, no. 4, pp. 623–644, 2000.

[49] N. Alon and J. H. Spencer, *The Probabilistic Method*. John Wiley & Sons, 2004.

[50] M. Ajtai, V. Chvátal, M. M. Newborn, and E. Szemerédi, "Crossing-free subgraphs," *Ann. Dis. Math.*, vol. 12, pp. 9–12, 1982.

[51] L. A. Székely, "Short proof for a theorem of Pach, Spencer, and Toth," *Contemporary Mathematics*, vol. 342, pp. 281–283, 2004.

[52] A. Fernandez and K. Efe, "Efficient VLSI layouts for homogeneous product networks," *IEEE Trans. Comput.*, vol. 46, no. 10, pp. 1070–1082, Oct. 1997.

[53] L. Sassatelli, S. K. Chilappagari, B. Vasic, and D. Declercq, "Two-bit message passing decoders for LDPC codes over the binary symmetric channel," Dec. 2009, http://arxiv.org/abs/0901.2090.

[54] Synopsys Inc., "32/28nm generic library," https://www.synopsys.com/Community/UniversityProgram/Pages/32-28nm-generic-library.aspx, 2015.

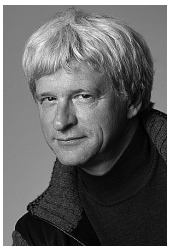
[55] K. Ganesan, "LDPC decoding power models," http://web.stanford.edu/~karthik3/JSACPowerModels/, 2015.

[56] K. Cho and D. Yoon, "On the general BER expression of one and two-dimensional amplitude modulations," *IEEE Trans. Comm.*, vol. 50, no. 7, pp. 1074–1080, July 2002.
[57] Y. Wang, J. S. Yedidia, and S. C. Draper, "Construction of high-girth QC-LDPC codes," Mitsubishi Electric Research Lab, Tech. Rep. TR2008-061, Sept. 2008.
[58] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. J.*, vol. 27, pp. 379–423,623–656, Jul./Oct. 1948.
[59] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
[60] I. Jacobs and E. Berlekamp, "A lower bound to the distribution of computation for sequential decoding," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 167–174, Apr. 1967.
[61] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.
[62] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. VLSI*, vol. 11, pp. 976–996, Dec. 2003.
[63] M. Mohiyuddin, A. Prakash, A. Aziz, and W. Wolf, "Synthesizing interconnect-efficient low density parity check codes," in *Proc. of 41st DAC*, June 2004, pp. 488–491.
[64] J. Thorpe, "Design of LDPC graphs for hardware implementation," in *Proc. of ISIT*, Lausanne, Switzerland, July 2002.
[65] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Comm.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
[66] J. Pach, F. Shahrokhi, and M. Szegedy, "Applications of the crossing number," *Algorithmica*, vol. 16, no. 1, pp. 111–117, 1996.
[67] S. Lin and D. J. Costello, *Error control coding*. Prentice Hall, 2004.
[68] H. Jeong and P. Grover, "Energy-adaptive codes," in *53rd Allerton Conference*, Monticello, IL, Oct. 2015.
[69] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite length codes," in *Proc. of 3rd ISTC*, Brest, France, 2003.
[70] C. Desset, M. Benoit, and L. Vandendorpe, "Computing the word-, symbol-, and bit-error rates for block error-correcting codes," *IEEE Trans. Comm.*, vol. 52, no. 6, pp. 910–921, June 2004.
[71] R. M. Corless, D. J. Jeffrey, and D. E. Knuth, "A sequence of series for the Lambert W function," *Proc. of ISSAC*, pp. 197–204, 1997.
[72] A. Hoorfar and M. Hassani, "Inequalities on the Lambert W function and hyperpower function," *J. Inequal. Pure and Appl. Math*, vol. 9, no. 2, 2008.

Karthik Ganesan received the B.S. degree in EECS and the B.A. degree in Statistics from the University of California at Berkeley in 2013, and the M.S. degree in EE from Stanford University in 2015, where he is currently pursuing the Ph.D. degree. He is interested in some aspects of coding theory, applied probability and ergodic theory, and their uses in fault-tolerant computing, low-power system design, and emerging neuroscience applications.

Pulkit Grover is an assistant professor in Electrical and Computer Engineering at Carnegie Mellon University (since 2013), working on information theory, circuit design, and biomedical engineering. His focus is on developing a new theory of information for low-energy communication, sensing, and computing by incorporating novel circuit/processing-energy models to add to classical communication or sensing energy models. A common theme in his work is observing when optimal designs depart radically from classical theoretical intuition. To apply these ideas to a variety of problems including wearables, IoT, and novel biomedical systems, his lab works extensively with engineers, neuroscientists, and doctors.

He is a recipient of the 2010 best student paper award at the IEEE Conference on Decision and Control; a 2010 best student paper finalist at the IEEE International Symposium on Information Theory; the 2011 Eli Jury Dissertation Award from UC Berkeley; the 2012 Leonard G. Abraham award from the IEEE Communications Society; a 2014 best paper award at the International Symposium on Integrated Circuits; a 2014 NSF CAREER award; and a 2015 Google Research Award.



Jan Rabaey is the Donald O. Pederson Distinguished Professor in the Electrical Engineering and Computer Science Department, University of California at Berkeley. He is currently the Scientific Co-director of the Berkeley Wireless Research Center (BWRC), the director of the Berkeley Ubiquitous SwarmLab, and the Director of the FCRP Multiscale Systems Research Center (MuSyC). His research interests include the conception and implementation of next-generation integrated wireless systems.

Dr. Rabaey is the recipient of a wide range of awards, among which are the 2008 IEEE Circuits and Systems Society Mac Van Valkenburg Award and the 2009 European Design Automation Association (EDAA) Lifetime Achievement award. In 2010, he was awarded the prestigious Semiconductor Industry Association (SIA) University Researcher Award. He is an IEEE Fellow and a member of the Royal Flemish Academy of Sciences and Arts of Belgium. He received his Ph.D. degree in applied sciences from the Katholieke Universiteit Leuven, Leuven, Belgium.

Andrea Goldsmith is the Stephen Harris professor in the School of Engineering and a professor of Electrical Engineering at Stanford University. She was previously on the faculty of Electrical Engineering at Caltech. Her research interests are in information theory and communication theory, and their application to wireless communications and related fields. She co-founded and served as Chief Scientist of Wildfire.Exchange, and previously co-founded and served as CTO of Quantenna Communications, Inc. She has also held industry positions at Maxim Technologies, Memorylink Corporation, and AT&T Bell Laboratories. Dr. Goldsmith is a Fellow of the IEEE and of Stanford, and has received several awards for her work, including the IEEE ComSoc Edwin H. Armstrong Achievement Award as well as Technical Achievement Awards in Communications Theory and in Wireless Communications, the National Academy of Engineering Gilbreth Lecture Award, the IEEE ComSoc and Information Theory Society Joint Paper Award, the IEEE ComSoc Best Tutorial Paper Award, the Alfred P. Sloan Fellowship, the WICE Technical Achievement Award, and the Silicon Valley/San Jose Business Journal's Women of Influence Award. She is author of the book "Wireless Communications" and co-author of the books "MIMO Wireless Communications" and "Principles of Cognitive Radio," all published by Cambridge University Press, as well as an inventor on 28 patents. She received the B.S., M.S. and Ph.D. degrees in Electrical Engineering from U.C. Berkeley.

Dr. Goldsmith has served on the Steering Committee for the IEEE Transactions on Wireless Communications and as editor for the IEEE Transactions on Information Theory, the Journal on Foundations and Trends in Communications and Information Theory and in Networks, the IEEE Transactions on Communications, and the IEEE Wireless Communications Magazine. She participates actively in committees and conference organization for the IEEE Information Theory and Communications Societies and has served on the Board of Governors for both societies. She has also been a Distinguished Lecturer for both societies, served as President of the IEEE Information Theory Society in 2009, founded and chaired the student committee of the IEEE Information Theory society, and chaired the Emerging Technology Committee of the IEEE Communications Society. At Stanford she received the inaugural University Postdoc Mentoring Award, served as Chair of Stanford's Faculty Senate in 2009 and currently serves on its Faculty Senate, Budget Group, and Task Force on Women and Leadership.